3. SYSTEM ARCHITECTURE

3.1. OVERVIEW AND MEMORY MAPS

3.1.1. Block Diagram

A block diagram of the MCU is shown in Figure 3-1.

3.1.2. Top Level Memory Map

The top level memory map in Figures 3-2 shows major boundaries between I/O, internal ROM, internal RAM, and external addresses. The individual registers are identified in Table 3-1.

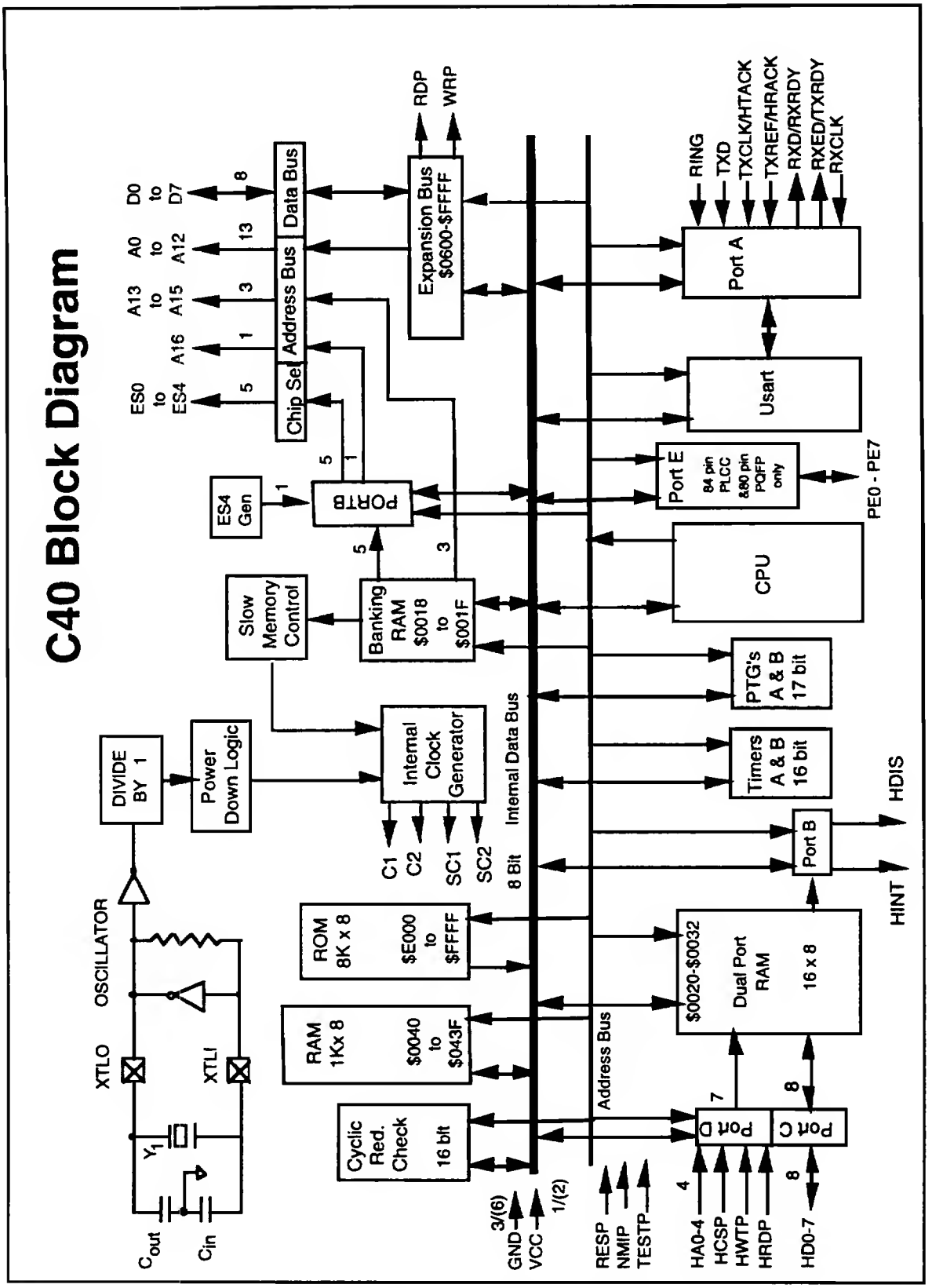


Figure 3-1a. MCU Block Diagram - C40

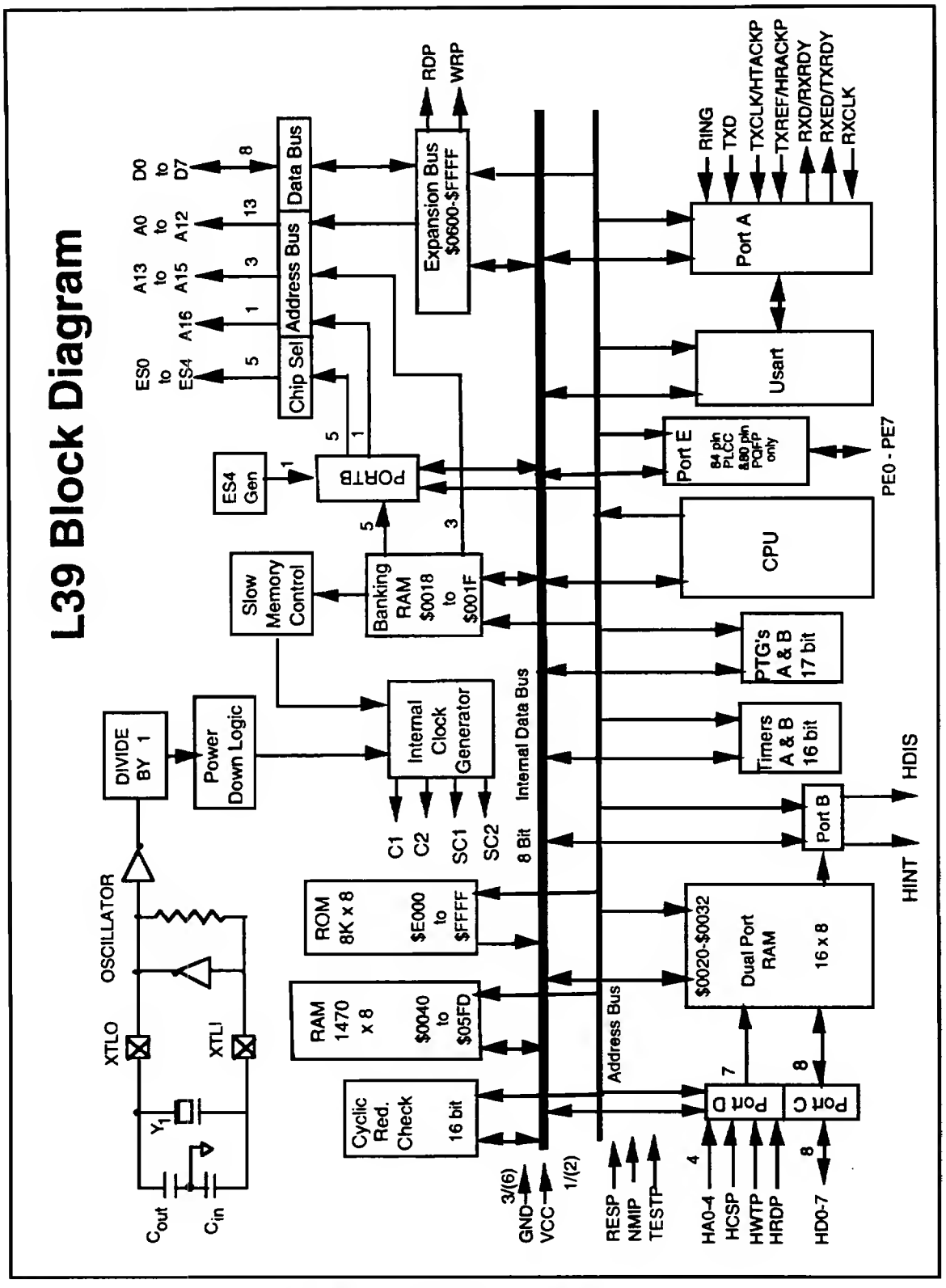


Figure 3-1b. MCU Block Diagram - L39

Bank Select Register	Address (Hex)	
	FFE0 - FFFF	Interrupt/JSR Vectors (32)
BSR7 (8K)	FFDF	
	F000	Internal ROM (8K-32), TSTP High
	E000	Default ESS Expansion, TSTP Low
	DFFF	
BSR6 (8K)	D000	Default
	C000	ES3 Expansion (32K)
	BFFF	
BSR5 (8K)	B000	
	A000	
	9FFF	
BSR4 (8K)	9000	
	8000	
	7FFF	
BSR3 (8K)	7000	Default
	6000	ES1 Expansion (16K)
	5FFF	
BSR2 (8K)	5000	
	4000	
	3FFF	Default
BSR1 (8K)	3000	ES1 Expansion (8K)
	2000	
	17FF	Default
BSR0 (8K)	1000	ES0 Expansion (6K)
	0800	
	0600 - 07FF	ES4 (512)
	0000 - 05FF	RAM and Registers (1536)

Figure 3-2a. MCU Memory Map Overview - 0 - FFFF

Addr. (Hex)	C40
07FF 0700	Page 7 (256) Available Externally (Part of ES4 when PBS1 = 0)
06FF 0600	Page 6 (256) Available Externally (Part of ES4 when PBS1 = 0)
05FE - 05FF	CRC Control (2)
  0500	Page 5 Unavailable (254)
0440-04FF	Page 4 Unavailable (192)
043F 0400	Page 4 RAM (64)
03FF 0300	Page 3 RAM (256)
02FF 0200	Page 2 RAM (256)
01FF 0100	Page 1 RAM (256)
00FF 0080	Page 0 RAM (128)
007F 0040	Page 0 RAM (64)
0038 - 003F	USART Registers (8)
0034 - 0037	Prec. Time Generator A (4)
0033	CS Speed Select
0020 - 0032	16550 I/F & DP RAM (19)
0018 - 00FF	Bank Select Registers (8)
0010 - 0017	Timer/Counters A &B (8)
000C - 000F	Prec. Time Generator B (4)
0000 - 000B	Ports A to E and Misc. Registers (12)

Addr. (Hex)	L39
07FF 0700	Page 7 (256) Available Externally (Part of ES4 when PBS1 = 0)
06FF 0600	Page 6 (256) Available Externally (Part of ES4 when PBS1 = 0)
05FE - 05FF	CRC Control (2)
05FD 0500	Page 5 RAM (254)
04FF 0400	Page 4 RAM (256)
03FF 0300	Page 3 RAM (256)
02FF 0200	Page 2 RAM (256)
01FF 0100	Page 1 RAM (256)
00FF 0080	Page 0 RAM (128)
007F 0040	Page 0 RAM (64)
0038 - 003F	USART Registers (8)
0034 - 0037	Prec. Time Generator A (4)
0033	CS Speed Select
0020 - 0032	16550 I/F & DP RAM (19)
0018 - 00FF	Bank Select Registers (8)
0010 - 0017	Timer/Counters A &B (8)
000C - 000F	Prec. Time Generator B (4)
0000 - 000B	Ports A to E and Misc. Registers (12)

Figure 3-2b. MCU Memory Map Overview - 0 - 07FF

Table 3-1a. Memory Map: 0-001Fh

Address	Read	Write
0000	Port A	Port A
0001	--	Port B (Output Only)
0002	Port C	Port C
0003	Port D	Port D (0-3), Direction (4-7)
0004	--	Port A Direction
0005	Port B Select	Port B Select
0006	--	Port C Direction
0007	Port E	Port E
0008	Mask Option Register	Port E Direction
0009	Low Power Register (LPR)	Low Power Register (LPR)
000A	External Interrupt Register (EIR)	External Interrupt Register (EIR)
000B	Clear External Interrupt	Clear External Interrupt
000C	PTG B Mode (PBM)	PTG B Mode (PBM)
000D	PBB	PBB
000E	PBUL	PBUL, PBB to PBLL
000F	PBUL, Clear Flag	PBUL,PBB to PBLL, Download, Clear Flag
0010	Timer A Mode (TAM)	Timer A Mode (TAM)
0011	TALC, TAUC to TAS	TALL
0012	TAS	TAUL
0013	TAS, Clear Flag	TAUL, Download, Clear Flag
0014	Timer B Mode (TBM)	Timer B Mode (TBM)
0015	TBLC, TBUC to TBS	TBLL
0016	TBS	TBUL
0017	TBS, Clear Flag	TBUL, Download, Clear Flag
0018	Bank 0	Bank 0
0019	Bank 1	Bank 1
001A	Bank 2	Bank 2
001B	Bank 3	Bank 3
001C	Bank 4	Bank 4
001D	Bank 5	Bank 5
001E	Bank 6	Bank 6
001F	Bank 7	Bank 7

Table 3-1b. Memory Map: 0020h-0032h

Address	Host 16550A Emulation Mode		Host General Purpose Mode	
	Read	Write	Read	Write
0020	TX FIFO Buffer	RX FIFO Buffar	TX FIFO Buffer	RX FIFO Buffer
0021	Line Status Register (LSR)	LSR1 Only	SP RAM 1	SP RAM 1
0022	Modem Status Register MSR)	Modem Status Register (MSR)	SP RAM 2	SP RAM 2
0023	Line Control Register (LCR)	*	SP RAM 3	SP RAM 3
0024	Modem Control Register (MCR)	*	SP RAM 4	SP RAM 4
0025	FIFO Control Register (FCR)	FIFO Control Register (FCR)	SP RAM 5	SP RAM 5
0026	SP RAM 6	SP RAM 6	SP RAM 6	SP RAM 6
0027	SP RAM 7	SP RAM 7	SP RAM 7	SP RAM 7
0028	Divisor Latch LSB	Divisor Latch LSB	SP RAM 8	SP RAM 8
0029	Divisor Latch MSB	Divisor Latch MSB	SP RAM 9	SP RAM 9
002A	SP RAM A	SP RAM A	SP RAM A	SP RAM A
002B	SP RAM B	SP RAM B	SP RAM B	SP RAM B
002C	SP RAM C	SP RAM C	SP RAM C	SP RAM C
002D	SP RAM D	SP RAM D	SP RAM D	SP RAM D
002E	GP DP FIFO Status (GPFS)	GPFS3 Only	GP FIFO Status (GPFS)	*
002F	Host Handshake Register (HHR)	Host Handshake Register (HHR)	Host Handshake Register (HHR)	Host Handshake Register (HHR)
0030	FIFO Status Register (FSR)	FIFO Status Registar (FSR)	FIFO Status Register (FSR)	FIFO Status Register (FSR)
0031	FIFO Interrupt Enable Register (FIER)	FIFO Interrupt Enable Register (FIER)	FIFO Interrupt Enable Register (FIER)	FIFO Interrupt Enable Register (FIER)
0032	Host Control Register (HCR)	Host Control Registar (HCR)	Host Control Register (HCR)	Host Control Register (HCR)

Table 3-1c. Memory Map: 0033h-003Fh

Address	Read	Write
0033	Chip Select Fast/Slow	Chip Select Fast/Slow
0034	PTG A Mode (PAM)	PTG A Mode (PAM)
0035	PAB	PAB
0036	PAUL	PAUL, PAB to PALL
0037	PAUL, Clear Flag	PAUL,PAB to PALL, Download, Clear Flag
0038	Serial In Buffer (SIB)	Serial Out Buffer (SOB)
0039	Sarial Interrupt Enable (SIR)	Serial Interrupt Enable (SIR)
003A	Serial Mode Register (SMR)	Serial Mode Register (SMR)
003B	Serial Line Control Register (SLCR)	Serial Line Control Register (SLCR)
003C	Serial Status Register (SSR)	Serial Status Register (SSR)
003D	Serial Form Register (SFR)	Serial Form Register (SFR)
003E	--	SOUT (RXD) Divider Latch (SODL)
003F	--	SIN (TXD) Divider Latch (SIDL)

Table 3-1d. Memory Map: 0040h–07FFh (C40)

Address	Read	Write
0040 . 00FF	Page 0 RAM (192 bytes)	
0100 . 01FF	Page 1 RAM (256 bytes)	
0200 . 02FF	Page 2 RAM (256 bytes)	
0300 . 03FF	Page 3 RAM (256 bytes)	
0400 . 043F	Page 4 RAM (64 bytes)	
0440 . 04FF	Page 4 Not Available (192 bytes)	
0500 . 05FD	Page 5 Not Available (254 bytes)	
05FE	CRC-L	CRC INPUT BUFFER
05FF	CRC-H	INITIALIZE CRC
0600 . 07FF	ES4 Active (PBS1 = 0) (512 bytes)	ES4 Inactive (PBS1 = 1) (512 bytes)
0800 . 1FFF	Bank Select 0 (6K PBS1 = 0 or 6.5K PBS1 = 1)	

Table 3-1e. Memory Map: 0040h–07FFh (L39)

Address	Read	Write
0040 . 00FF	Page 0 RAM (192 bytes)	
0100 . 01FF	Page 1 RAM (256 bytes)	
0200 . 02FF	Page 2 RAM (256 bytes)	
0300 . 03FF	Page 3 RAM (256 bytes)	
0400 . 04FF	Page 4 RAM (256 bytes)	
0500 . 05FD	Page 5 RAM (254 bytes)	
05FE	CRC-L	CRC INPUT BUFFER
05FF	CRC-H	INITIALIZE CRC
0600 . 07FF	ES4 Active (PBS1 = 0) (512 bytes)	ES4 Inactive (PBS1 = 1) (512 bytes)
0800 . 1FFF	Bank Select 0 (6K PBS1 = 0 or 6.5K PBS1 = 1)	

3.1.3. I/O Register Bit Assignments

The individual I/O signal and data bits are identified in Table 3-2.

Table 3-2a. Register Bit Assignments: 0000h - 0017h

Addr.	Function	Bit 7	6	5	4	3	2	1	0
0000	Port A Data (0-7; I/O)	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
0001	Port B Data (0-7; Output Only)	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
0002	Port C Data (0-7; I/O)	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
0003	Port D Data (0-3; I/O), Port D Direction (4-7)	PDD3	PDD2	PDD1	PDD0	PD3	PD2	PD1	PD0
0004	Port C Direction (0-7)	PAD7	PAD6	PAD5	PAD4	PAD3	PAD2	PAD1	PAD0
0005	Port B Select (0-7)	PBS7	PBS6	PBS5	PBS4	PBS3	PBS2	PBS1	PBS0
0006	Port C Direction (0-7)	PCD7	PCD6	PCD5	PCD4	PCD3	PCD2	PCD1	PCD0
0007	Port E Data (0-7)	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
0008	Mask Option Register (MSR)	Mask Option Bit 7 (MOR7)	Mask Option Bit 6 (MOR6)	Mask Option Bit 5 (MOR5)	Mask Option Bit 4 (MOR4)	Mask Option Bit 3 (MOR3)	Mask Option Bit 2 (MOR2)	Mask Option Bit 1 (MOR1)	Mask Option Bit 0 (MOR 0)
0009	Low Power Register (LPR)	Low Power Mode Enable	Low Power Mode	Wake up Enable PA0 (Ring)	Wake up Enable PD4 (Wrt) & PD5 (CS)	Wake up Enable PA2 (TXD)	Wake up Enable PD4 (DTR)	Wake up Enable PD5 (AL)	Not Used
000A	External Interrupt Register (EIR)	External Interrupt Flag PA1	External Interrupt Flag PA4	External Interrupt Flag PD7	External Interrupt Enable PA1	External Interrupt Enable PA4	External Interrupt Enable PD7	Edge Detect Polarity PA4	Edge Detect Polarity PD7
000B	Clear Interrupt Register (CIR)	Clear Interrupt Flag PA1	Clear Interrupt Flag PA4	Clear Interrupt Flag PD7	Clear Interrupt Flag PA7	Clear Interrupt Flag PA3	ES4 Fast Memory Cycle	Not Used	Not Used
000C	Precision Time Generator B (PTGB) Mode (PBM)	PTG B Interrupt Flag	PTG B Interrupt Enable	Not Used					PTG B Timer Mode
0010	Timer A Mode (TAM)	Timer A Interrupt Flag	Timer A Interrupt Enable	Timer A Vector Select	Not Used		Timer A Div by 32 Prescale	Timer A Mode Bit 1 (TAM1)	Timer A Mode Bit 0 (TAM0)
0011	Timer A Lower Latch	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0012	Timer A Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0013	Timer A Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0014	Timer B Mode (TBM)	Timer B Interrupt Flag	Timer B Interrupt Enable	Timer B Vector Select	Not Used		Timer B Div by 32 Prescale	0	0
0015	Timer B Latch Low	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0016	Timer B Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0017	Timer B Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8

Table 3-2b. Register Bit Assignments: 0018h - 001Fh

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0018	Bank Select Register 0 (BSR0)	ES3 (0)	ES2 (0)	ES1 (0)	ES0 (0)	A16 (0)	A15 (0)	A14 (0)	A13 (0)
0019	Bank Select Register 1 (BSR1)	ES3 (1)	ES2 (1)	ES1 (1)	ES0 (1)	A16 (1)	A15 (1)	A14 (1)	A13 (1)
001A	Bank Select Register 2 (BSR2)	ES3 (2)	ES2 (2)	ES1 (2)	ES0 (2)	A16 (2)	A15 (2)	A14 (2)	A13 (2)
001B	Bank Select Register 3 (BSR3)	ES3 (3)	ES2 (3)	ES1 (3)	ES0 (3)	A16 (3)	A15 (3)	A14 (3)	A13 (3)
001C	Bank Select Register 4 (BSR4)	ES3 (4)	ES2 (4)	ES1 (4)	ES0 (4)	A16 (4)	A15 (4)	A14 (4)	A13 (4)
001D	Bank Select Register 5 (BSR5)	ES3 (5)	ES2 (5)	ES1 (5)	ES0 (5)	A16 (5)	A15 (5)	A14 (5)	A13 (5)
001E	Bank Select Register 6 (BSR6)	ES3 (6)	ES2 (6)	ES1 (6)	ES0 (6)	A16 (6)	A15 (6)	A14 (6)	A13 (6)
001F	Bank Select Register 7 (BSR7)	ES3 (7)	ES2 (7)	ES1 (7)	ES0 (7)	A16 (7)	A15 (7)	A14 (7)	A13 (7)

Table 3-2c. Register Bit Assignments: 0020h - 0032h

Addr.	Function	Bit 7	6	5	4	3	2	1	0
0020	RX FIFO Buffer (Host Read/MCU Write)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0020	TX FIFO Buffer (Host Write/MCU Read)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0021	Line Status Register (LSR)	RX FIFO Error	TX FIFO Empty (TEMT)	TX FIFO Empty (THRE)	Break Interrupt (BI)	Framing Error (FE)	Parity Error (PE)	Overrun Error (OE)	RX Data Ready (DR)
0022	Modem Status Register (MSR)	Data Carrier Detect (DCD)	Ring Indicator (RI)	Data Set Ready (DSR)	Clear to Send (CTS)	Delta Data Carrier Detect (DDCD)	Trailing Edge of Ring Indicator (TERI)	Delta Data Set Ready (DDSR)	Delta Clear to Send (DCTS)
0023	Line Control Register (LCR)	DLAB	Set Break	Stick Parity	Even Parity	Parity Enable	Number Stop Bits	Word Length (WLS1)	Word Length (WLS0)
0024	Modem Control Register (MCR)	0	0	0	Loop	Out 2	Out 1	Request to Send (RTS)	Data Terminal Ready (DTR)
0025	FIFO Control Register (FCR)	RCVR Trigger MSB	RCVR Trigger LSB	Reserved	Reserved	DMA Mode Select	TX FIFO Reset	RX FIFO Reset	FIFO Enable
Host Rd/Wt @ x1	Interrupt Enable Register (IER)	0	0	0	0	Modem Status Interrupt Enable (EDSSI)	RX Line Status Interrupt Enable (ELSI)	TX Hold Empty Interrupt Enable (ETBEI)	RX Data Avail Interrupt Enable (ERBFI)
Host Read @ x2	Interrupt Identifier Register (IIR)	FIFO Enable (FCR0)	FIFO Enable (FCR0)	0	0	Interrupt ID Bit 2	Interrupt ID Bit 1	Interrupt ID Bit 0	0 If Interrupt Pending
0027	Scratch Register (SCR)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0028	Divisor Latch LSB	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0029	Divisor Latch MSB	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
002E	GP FIFO Status (GPFS)	Tx Empty Interrupt Enable	Tx Empty Interrupt Flag	Tx FIFO Half Empty	Rx Trig Level Sel (RTL1)	Rx Trig Level Sel (RTL0) [GP Mde] TuClk Off [16550]	Rx Trig Level Int Enable	Rx Trig Level Int Flag	RCNE [GP Mde] Rx FIFO Data Avail [16550]
002F	Host Handshake Register (HHR)	Controller Intrpt 1 Flag (CF1)	Controller Intrpt 2 Flag (CF2)	Host Intrpt 1 Flag (HF1)	Host Intrpt 2 Flag (HF2)	Controller Intrpt 1 Enable (CE1)	Controller Intrpt 2 Enable (CE2)	Host Intrpt 1 Enable (HE1)	Host Intrpt 2 Enable (HE2)
0030	FIFO Status Register (FSR)	TX FIFO Half Full Flag (TCHF)	Data Avail (TCDA)	RX FIFO Freeze	Receiver Break Interrupt (BI)	Receiver Framing Error (FE)	Receiver Parity Error (PE)	RX FIFO Empty Flag (RCEMT)	RX FIFO Half Emty Flag (RCHE)
0031	FIFO Interrupt Enable Register (FIER)	TX FIFO Half Full Interrupt Enable (TCHFE)	Data Avail Interrupt Enable (TCDAE)	UART Timing Select	RUCLK Off	TX FIFO Reset	RX FIFO Reset	RX FIFO Empty Interrupt Enable (RCEMTE)	RX FIFO Half Emty Interrupt Enable (RCHEE)
0032	Host Control Register (HCR)	TX FIFO Full Interrupt Flag	MCR Write Flag	LCR Write Flag	Divisor Latch Write Flag	RX FIFO Interrupt	Host Mode Select	16450/ 16550 Mode	16450/ 16550 Interrupt Enable

Table 3-2d. Register Bit Assignments: 0033h - 003Fh

Addr.	Function	Bit 7	6	5	4	3	2	1	0
0033	ES Speed (ESS)	ES3-1	ES3-0	ES2-1	ES2-0	ES1-1	ES1-0	ES0-1	ES0-0
0034	Precision Time Generator A (PTGA) Mode (PAM)	PTG A Interrupt Flag	PTG A Interrupt Enable	Not Used				SIN 0 = PTGB 1 = PTGA	PTGA Timer Mode
0038	Serial In Buffer (SIB) / Serial Out Buffer (SOB)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0039	Serial Interrupt Register (SIR)	TXD Status Interrupt Flag	TXCLK (PA3) ↑ Interrupt Flag	RXCLK (PA7) ↓ Interrupt Flag	TXCLK (PA3)↑ Interrupt Enable	RXCLK (PA7) ↓ Interrupt Enable	TXD Status Interrupt Enable	RXD Buf Empty Interrupt Enable (BE)	RXD Buf Full Interrupt Enable (BF)
003A	Serial Mode Register (SMR)	RXD On	TXD On	Timing Select 0 = PRGs 1 = TIMB	Sync Mode	TXD Sync Bit	TXREF Clock Select	CTSP/ RLSDP Sync Enable	Not Used
003B	Serial Line Control Register (SLC)	Parity Stuff Bit	Set Break	Stuff Parity	Even Parity	Enable Parity	Two Stop Bits	Word Length Bit 1 (SL1)	Word Length Bit 0 (SL0)
003C	Serial Status Register (SSR)	TXD Parity Bit	RXD Underrun (UR)	RXD Buffer Empty (BE)	TXD Break Int (BI)	TXD Framing Error (FE)	TXD Parity Error (PE)	TXD Overrun Error (OE)	TXD Buffer Full (BF)
003D	Serial Form Register (SFR)	TXD/ TXDE Echo	REXD/ TXD Echo	TXD/ RXD Echo	TIMA Input TXD Test	TXD/ Edge TXD Test	PTG Divide by 3	7/8 Short Stop Bit	3/4 Short Stop Bit
003E	SOUT (RXD) Divider Latch (SODL)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
003F	SIN (TXD) Divider Latch (SIDL)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0

3.2. CENTRAL PROCESSING UNIT (CPU)

The central processing unit (CPU) is an enhanced 8-bit 6502 CPU. The CPU executes stored instructions fetched from memory (usually internal masked ROM) sequentially unless a jump to a new location is specified in the instruction or an interrupt occurs. Operation of the CPU instructions are described in Appendix A. The CPU is 6502 instruction compatible except "(indirect, X)" addressing mode changed to "(indirect)," and "(indirect), Y" changed to "(indirect), X".

The MCU CPU registers are the same as the 6502 CPU with the addition of the W-register and I-register.

The data flow for the CPU registers is illustrated in Figure 3-3.

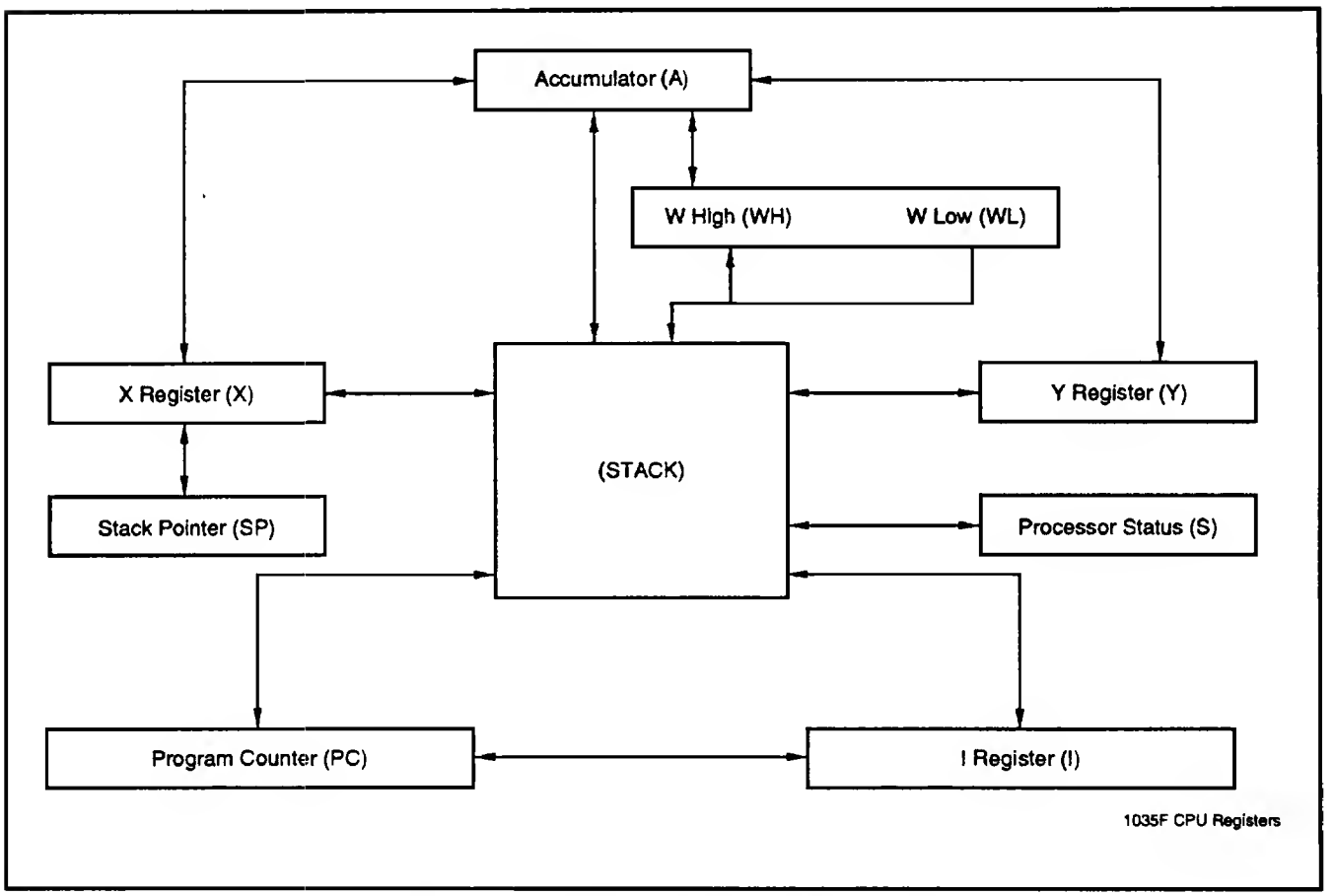


Figure 3-3. CPU Registers and Data Flow

3.2.1. Index Registers

There are two 8-bit index registers: X and Y. Either index register can be used as a base to modify the program counter contents and thus obtain a new address—the sum of the program counter contents and the index register contents. When executing an instruction which specifies indexed addressing, the CPU fetches the op code and the address, and modifies the address from memory by adding the index register to it prior to loading or storing the value of memory.

3.2.2. Stack Pointer

The Stack Pointer is an 8-bit register that controls access to the stack. The stack is initialized under software control usually to the top of Page 1 RAM. The stack length can be up to 256 bytes ($1FF down to $100).

The Stack Pointer is automatically incremented and decremented under control of the CPU to perform stack manipulation in response to program instructions, a reset, a non-maskable interrupt (NMI), an internally generated interrupt request (IRQ), or execution of the CPU Break (BRK) instruction. The Stack Pointer must be initialized by the user program. The JSR, JPI, PIA, BRK, RTI, and RTS instructions use the stack and the Stack Pointer.

3.2.3. Arithmetic and Logic Unit (ALU)

All arithmetic and logic operations take place in the ALU, including incrementing and decrementing internal registers (except the Program Counter). The ALU cannot store data for more than one cycle. If data is placed on the inputs to the ALU at the beginning of a cycle, the result is always gated into one of the storage registers or to memory during the next cycle.

Each bit of the ALU has two inputs. These inputs can be tied to various internal buses or to a logic 0; the ALU then generates the function (AND, OR, SUM, and so on) using the data on the two inputs.

3.2.4. Accumulator (A)

The Accumulator (A) is a general purpose 8-bit register that stores the results of most arithmetic and logic operations. In addition, the Accumulator usually contains one of the two data bytes used in these operations.

3.2.5. Program Counter (PC)

The 16-bit Program Counter (PC) provides the addresses that step the processor through sequential instructions in a program. Each time the processor fetches an instruction from the program memory, the least significant byte of the Program Counter (PCL) is placed on the eight low-order lines of the internal address bus and the most significant byte of the Program Counter (PCH) is placed on the eight high-order lines of the internal address bus. The Program Counter is incremented each time an instruction or data byte is fetched from program memory.

3.2.6. Instruction Register and Instruction Decode

Instructions are fetched from ROM or RAM and gated onto the internal data bus. These instructions are latched then decoded along with timing and interrupt signals to generate control signals for the various registers.

3.2.7. W Register (W)

The 16-bit W register is used exclusively to perform the accumulate function during execution of the Multiply Accumulate (MPA) instruction.

3.2.8. I Register (I)

The 16-bit I register is used for threaded code instructions. Note that the I register should not be confused with the CPU Instruction register which is not addressable.

3.2.9. Processor Status Register (PSR)

The 8-bit Processor Status Register (Table 3-3) contains seven status flags. Some of these flags are controlled by the user program; others may be controlled both by the user program and the CPU. The instruction set contains a number of conditional branch instructions which allow testing of these flags. Each of the seven processor status flags is described in the following paragraphs.

Table 3-3. Processor Status Register (PSR)

Function	Bit							
	7	6	5	4	3	2	1	0
Processor Status Register (PSR)	Negative (N)	Overflow (V)	Not Used	Break (B)	Decimal Mode (D)	IRQ Interrupt Disable (I)	Zero (Z)	Carry (C)

Bit 7: **Negative (N).** The Negative (N) bit copies the arithmetic sign bit value resulting from a data movement or an arithmetic operation. If the sign bit is set, the resulting values of the data movement or arithmetic operation is negative and the N bit is a logic 1; if the sign bit is cleared, the result of the data movement or arithmetic operation is positive and the N bit is a logic 0. There are no instructions that set or clear the N bit since the N bit represents only the status of a result. Standard 6502 instructions that affect the state of the N bit are: ADC, AND, ASL, BIT, CMP, CPX, CPY, DEC, DEX, DEY, EOR, INC, INX, INY, LDA, LDX, LDY, LSR, ORA, PLA, PLP, ROL, ROR, RTI, SBC, TAX, TAY, TSX, TXA and TYA. New instructions that affect this bit are: ADD, ASR, LAB, LAI, LAN, MPA, MPY, NEG, PIA, RND, TAW, and TWA.

1 = Negative value

0 = Positive value

Bit 6: **Overflow (V).** The Overflow (V) bit indicates that the result of a signed, binary addition or subtraction operation is a value that cannot be contained in seven bits ($-128 \leq n \leq +127$). The V bit only has meaning when signed arithmetic (sign and seven magnitude bits) is performed. When the ADC or SBC instruction is performed, the V bit is set to logic 1 if the polarity of the sign bit (bit 7) is changed because the result exceeds +127 or -128; otherwise the V bit is cleared to logic 0. The V bit may also be cleared under program control by the Clear Overflow (CLV) instruction. There is no instruction to set V.

The Overflow bit may also be used with the BIT instruction. The BIT instruction, which may be used to sample interface devices, allows the V bit to reflect the condition of bit 6 in the sampled field. During a BIT instruction, the V bit is set to equal to the content of bit 6 of the data tested with the BIT instruction. When used in this mode, the V bit has nothing to do with signed arithmetic, but is just another sense bit for the CPU. Standard 6502 Instructions which affect the V flag are ADC, BIT, CLV, PLP, RTI and SBC. New instructions that affect this bit are: ADD (binary mode only), CLW, MPA, MPY and RND.

1 = Overflow set

0 = Overflow cleared

Bit 5: **Not Used.**

Bit 4: **Break (B).** The Break (B) bit indicates the condition which caused the IRQ service routine to be entered. If the IRQ service routine was entered because the CPU executed a BRK instruction, the B bit is set to logic 1. If the IRQ routine was entered as the result of an IRQ occurrence, the B bit is cleared to logic 0. There are no instructions which can set or clear this bit.

1 = BRK instruction

0 = No BRK instruction

Bit 3: **Decimal Mode (D).** The Decimal Mode (D) bit controls the arithmetic mode of the CPU. When the D bit is a logic 1, the adder operates as a decimal adder. When this bit is a logic 0, the adder operates as a straight binary adder. The adder mode is controlled only by two instructions. The Set Decimal Mode (SED) instruction sets the D bit; the Clear Decimal Mode (CLD) instruction clears it. The PLP and RTI instructions also affect the D bit. The D bit is cleared by assertion of RESP thus initially establishing binary mode.

1 = Decimal mode selected

0 = Binary mode selected

Bit 2: **IRQ Interrupt Disable (I).** The Interrupt Disable (I) bit controls the servicing of the internal interrupt request (IRQ). If the I bit is a logic 0, the IRQ will be serviced. If the bit is a logic 1, the IRQ will be ignored. The CPU sets the I bit to logic 1 if the external NMIP, external RESP, or the internal IRQ input (with the I bit set to a logic 0) is asserted. The I bit is restored by the Pull Processor Status from Stack (PLP) instruction, or as the result of executing a Return from Interrupt (RTI) instruction (provided the Interrupt Disable bit was cleared prior to the interrupt). The Interrupt Disable bit may be set or cleared under program control using a Set Interrupt Disable (SEI) or a Clear Interrupt Disable (CLI) instruction, respectively.

1 = IRQ interrupt disabled

0 = IRQ interrupt enabled

Bit 1: **Zero (Z).** The Zero (Z) bit is set to logic 1 by the CPU during any data movement or by any calculation which sets all eight bits of the result to zero. This bit is cleared to logic 0 when the resultant eight bits of a data movement or calculation operation are not all zero. The instruction set contains no instruction to specifically set or clear the Zero bit. The Z bit is, however, affected by the following standard 6502 instructions: ADC, AND, ASL, BIT, CMP, CPX, CPY, DEC, DEX, DEY, EOR, INC, INX, INY, LDA, LDX, LDY, LSR, ORA, PLA, PLP, ROL, ROR, RTI, SBC, TAX, TAY, TXA, TSX and TYA. New instructions that affect this bit are: ADD, ASR, LAB, LAI, LAN, NEG, PIA, TAW, and TWA.

1 = Zero result

0 = Non-zero result

Bit 0: **Carry (C).** The Carry (C) bit can be considered as the ninth bit of an arithmetic operation. It is set to logic 1 if a carry from the eighth bit has occurred, or cleared to logic 0 of no carry occurred, as the result of arithmetic operations. The Carry bit may be set or cleared under program control by use of the Set Carry (SEC) or Clear Carry (CLC) instruction, respectively. Other standard 6502 operations which affect the C bit are ADC, ASL, CMP, CPX, CPY, LSR, PLP, ROL, ROR, RTI and SBC. New instructions that affect this bit are: ADD (binary mode only).

1 = Carry set

0 = Carry cleared

3.2.10. CPU Interrupt Logic

CPU interrupt logic controls the sequencing of the RESP, NMIP, and IRQ activated interrupts and the CPU BRK instruction.

RES Sequencing

A low-to-high transition on RESP causes the Interrupt Disable (I) bit in the Processor Status Register to be set and program execution to begin at the address fetched from the RES vector ($FFFE and $FFFF).

NMI Sequencing

At the first operation code fetch following the high-to-low transition of the NMIP input, the interrupt logic forces execution of the Break (BRK) instruction and subsequent execution from the address vector stored at $FFFC and $FFFD. Simultaneous with the execution of the BRK instruction, the Interrupt Disable bit in the Processor Status Register is set to disable an IRQ.

IRQ Sequencing

An IRQ interrupt occurs when the Interrupt Disable (I) bit of the Process Status Register is cleared (0) and IRQ has been asserted from the IRQ Interrupt Logic. Upon IRQ interruption, the BRK instruction is forced and subsequent program execution begins at the IRQ interrupt service subroutine location specified by the IRQ interrupt vector corresponding to the IRQ number (1-6). The IRQ vector is located in one of six locations in ROM ($FFF0-$FFFB) or one of four locations in RAM ($0102-$0103, $0106-$0107, $0108-$0109, or $010A-$010B). The page 1 RAM IRQ vectors are Timer A and Timer B options, respectively.

The I bit is set to inhibit further IRQ interruption until completion of the IRQ interrupt service subroutine, at which time the I bit is automatically cleared by the RTI instruction. The I bit can also be cleared under program control with the CLI instruction.

For each IRQ that has multiple sources of interruption, the IRQ service subroutine must determine the source of the interrupt by examining applicable interrupts flags. The interrupt flag causing the IRQ should also be cleared after processing the interrupt and before returning to the interrupted routine.

3.3. OSCILLATOR

The Clock Oscillator provides the basic timing signals used by the MCU internal circuits. The reference frequency can be supplied by either a parallel resonant crystal or a clock input. The input frequency is divided by 1 to generate the internal Ø2 clock. Typical Clock Oscillator input circuits are shown in Figure 3-4.

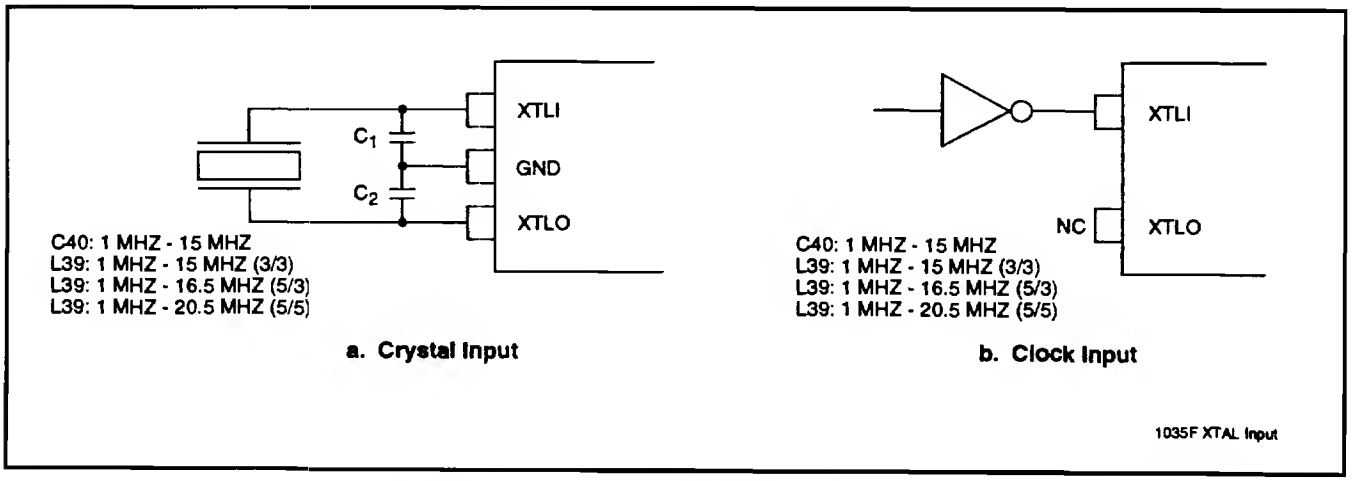


Figure 3-4. MCU Clock Oscillator Input Options

3.4. LOW POWER OPERATION

3.4.1. Low Power Mode (LPM)

The MCU provides a low power Sleep Mode and, in the L39 only, an ultra low power Stop Mode

3.4.2. Sleep Mode

The Sleep Mode provides low power consumption during periods of no host/modem interface activity and no telephone line activity while the modem internal clock is operating. In this mode, either host/modem or telephone line activity can be enabled to wake up the modem for immediate operation. Internal clocks are turned off but the internal oscillator is running.

3.4.3. Stop Mode (L39 Only)

The Stop Mode allows the host to completely turn off microcontroller operation except for wake up capability during periods of no modem operation as determined by the host. Internal clocks and the internal oscillator are turned off.

3.4.4. Wake-Up

Wake up can be enabled to occur upon a high being detected on PA0 (Ring), a low being detected on PD4 (DTR), a low being detected on both PD4 (Host Chip Select) and PD5 (Host Write Enable), or a low being detected on PA2 (TXD Start Bit). (The typical signal connected to the corresponding port is shown in parenthesis.) Also, asserting the NMIP input will cause the MCU to wake-up.

Wake-up from the Sleep Mode takes two clock cycles.

Wake-up from the Stop Mode takes several milliseconds. An additional twelve cycles must elapse following wake-up before the MCU can be put back in the low power mode.

3.4.5. Low Power Register (LPR)

The Low Power Register (LPR), located at $0009, controls the low power mode (LPM) enable, mode selection, and wake-up condition enables (Table 3-4). All bits are cleared to logic 0 upon reset. All bits can be written and read. However, neither bit 6 nor 7 can be read in the logic 1 state since the CPU will be in the Sleep or Stop mode. Figure 3-5 illustrates the MCU low power mode logic and timing.

Table 3-4. Register Bit Assignments - Low Power Register (LPR) - 0009h

Addr.	Function	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0009	Low Power Register (LPR)	Low Power Mode Enable	Low Power Mode 0 = Sleep 1 = Stop	Wake up Enable PA0 (Ring)	Wake up Enable PD4 (Wrt) & PD5 (CS)	Wake up Enable PA2 (TXD)	Wake up Enable PD4 (DTR)	Wake up Enable PD5 (AL)	Not Used

Bits 0: **Not Used.**

Bits 1: **Wake-Up Enable for PD5 (AL).** Control bit.

1 = Enables the CPU to wake up from low power mode when a low is detected on PD5.

0 = Disables the CPU to wake up from low power mode when a low is detected on PD5.

Bit 2: **Wake-Up Enable for PD4 (DTR).** Control bit.

1 = Enables the CPU to wake up from low power mode when a low is detected on PD4.

0 = Disables the CPU to wake up from low power mode when a low is detected on PD4.

Bit 3: **Wake-Up Enable for PA2 (TXD Start Bit).** Control bit.

1 = Enables the MCU to wake up from low power mode when a low is detected on PA2.

0 = Disables the MCU to wake up from low power mode when a low is detected on PA2.

Bit 4: **Wake-Up Enable for PD4 (MCU Chip Select) and PD5 (Host Bus Write).** Control bit.

1 = Enables the MCU to wake up from low power mode when a low is detected on both PD4 and PD5.

0 = Disables the MCU to wake up from low power mode when a low is detected on both PD4 and PD5.

Bit 5: **Wake Up Enable for PA0 (Ring).** Control bit.

1 = Enables the MCU to wake up from low power mode when a high is detected on PA0.

0 = Disables the MCU to wake up from low power mode when a high is detected on PA0.

Bit 6: **Low Power Mode.** Control bit.

1 = Invokes the Stop low power mode if low power mode is enabled (LPR7 = 1). **Note:** The Stop mode is not supported in the C40 MCU, which enters Sleep Mode rather than Stop Mode.

0 = Invokes the Sleep low power mode if low power mode is enabled (LPR7 = 1).

Bit 7: **Low Power Mode Enable.** Control bit.

1 = Enables the low power mode selected by LPR6 to be entered when low power conditions exist

0 = Disables entry into the low power mode.

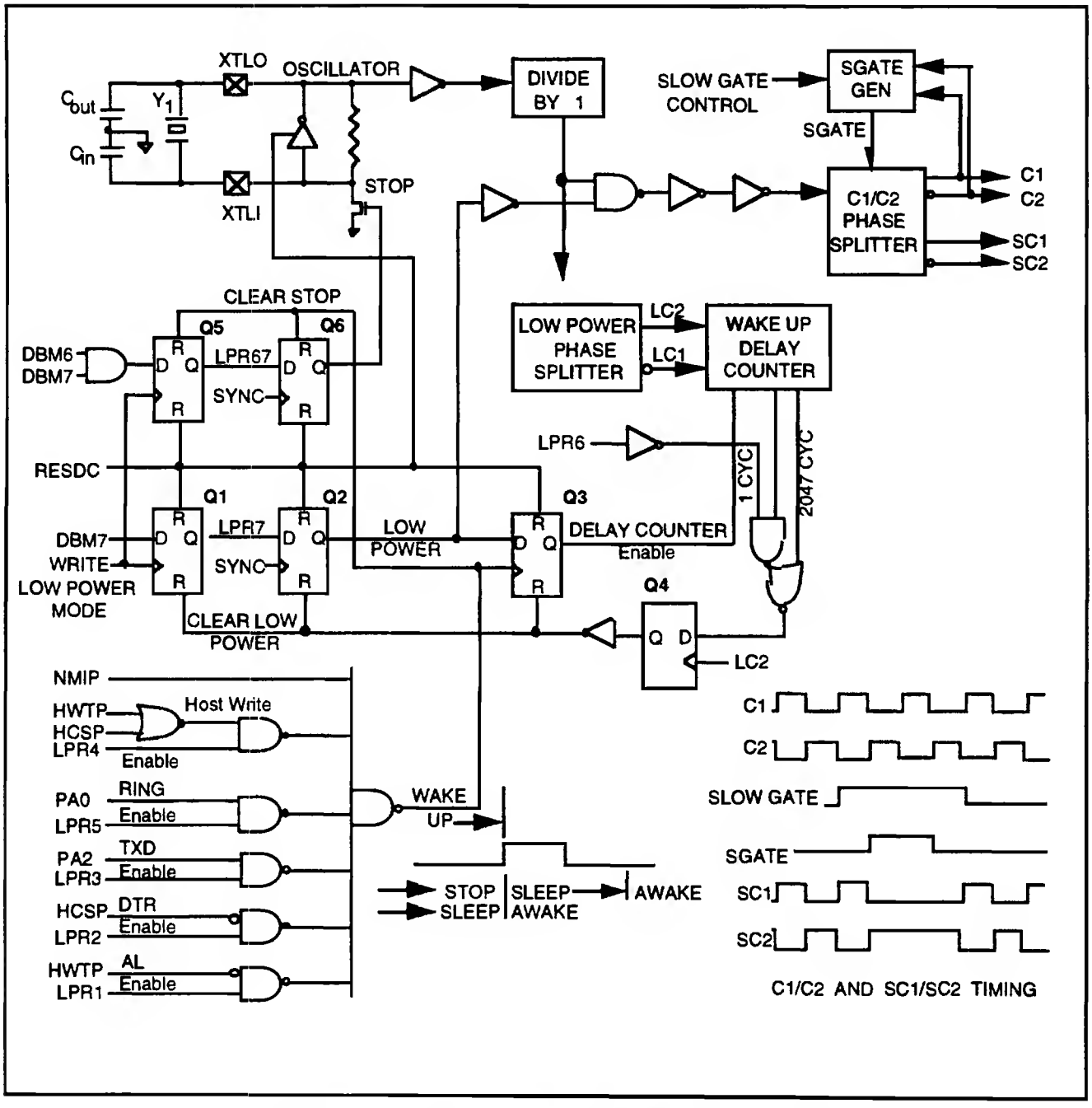


Figure 3-5. Low Power Mode Logic and Timing - L39

3.5. IRQ INTERRUPT LOGIC

3.5.1. Interrupt Request (IRQ)Vector and Hardware Priority

The IRQ Interrupt Logic prioritizes the individual interrupt requests (IRQ1-IRQ6) from the various sources and passes a single IRQ along with an IRQ number (1-6) and the IRQ vector page indicator to the CPU Interrupt Logic. Figure 3-6 illustrates the IRQ Interrupt Logic interface. Table 3-5 shows the IRQ interrupt levels, sources and vector addresses.

If simultaneous IRQs occur on IRQ1-IRQ6 lines, the number of the highest priority IRQ (1 = highest) is passed to the CPU. When the interrupt flag causing the IRQ is cleared by the IRQ interrupt service subroutine, the IRQ number of the highest pending IRQ is then passed.

The selection of ROM or RAM IRQ interrupt vectors for Timer A (IRQ5) and Timer B (IRQ3) is determined by bits 5 and 6 in the Timer A Mode and Timer B Mode registers, respectively (3.7).

3.5.2. Break Command

The BRK command causes the processor to go through an interrupt request sequence under program control. The address in the program counter (which points to the location of the BRK command + 1) is pushed on the stack, along with the processor status at the beginning of the BRK instruction. The processor then transfers control to the NMI interrupt vector (FFFC,D). Note that the BRK command cannot be masked by setting the I flag.

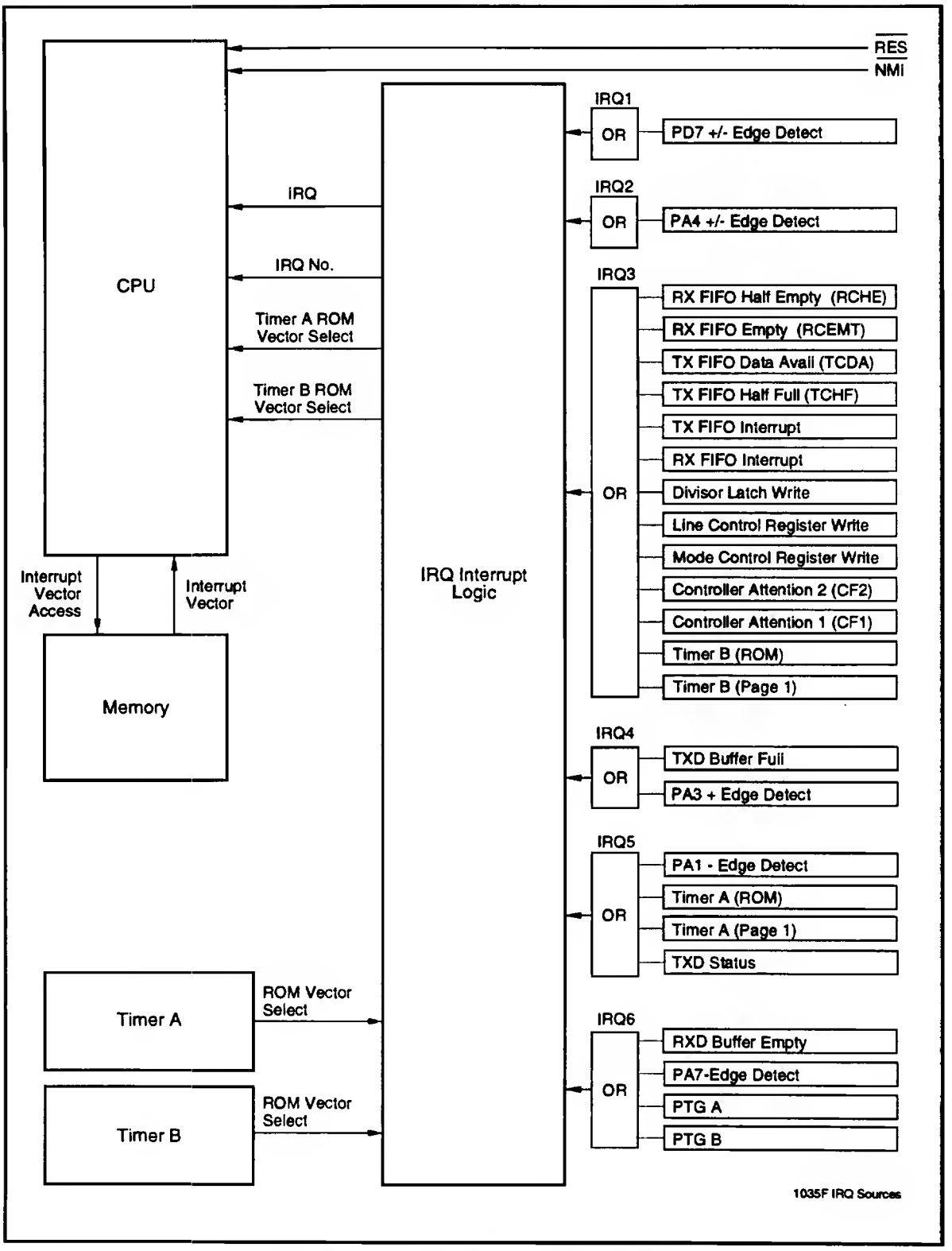


Figure 3-6. IRQ Interrupt Logic Interface

Table 3-5. Interrupt Request (IRQ)Vector and Hardware Priority

Source	Location		Interrupt Vector and Priority Level					
	Flag	Enable	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6
PD7 Edge Detect (+,-)	EIR5	EIR2	FFFA,B					
PA4 Edge Detect(+,-)	EIR6	EIR3		FFF8,9				
PA1 Edge Detect (-)	EIR7	EIR4					FFF2,3	
RX FIFO Half Empty (RCHE)	FSR0	FIER0			FFF6,7			
RX FIFO Empty (RCEMT)	FSR1	FIER1			FFF6,7			
TX FIFO Data Avail (TCDA)	FSR6	FIER6			FFF6,7			
TX FIFO Half Full (TCHF)	FSR7	FIER7			FFF6,7			
TX FIFO Interrupt	HCR7	FIER6, FIER7			FFF6,7			
RX FIFO Interrupt	HCR3	FIER0, FIER1			FFF6,7			
Divisor Latch Write	HCR4	HCR0			FFF6,7			
Line Control Register Write	HCR5	HCR0			FFF6,7			
Mode Control Register Write	HCR6	HCR0			FFF6,7			
Controller Attention 2 (CF2)	HHR6	HHR2			FFF6,7			
Controller Attention 1 (CF1)	HHR7	HHR3			FFF6,7			
Timer A (ROM)	TAM7	TAM6					FFF2,3	
Timer A (Page 1)	TAM6	TAM6					0102,3	
Timer B (ROM)	TBM7	TBM6			FFF6,7			
Timer B (Page 1)	TBM6	TBM6			0106,7			
TXD Buffer Full	SS0	SI0				FFF4,5		
TXD Status	SI7	SI2					FFF2,3	
PA3 Edge (+)	SI6	SI4				FFF4,5		
RXD Buffer Empty	SS5	SI1						FFF0,1
PA7 Edge Detect (-)	SI5	SI3						FFF0,1
PTGA	PAM7	PAM6						FFF0,1
PTGB	PBM7	PBM6						FFF0,1

3.6. INTERNAL ROM

The internal Read Only Memory (ROM) usually contains the user's program instructions and other fixed constants. These program instructions and constants are mask-programmed during fabrication.

3.6.1. Size and Location

The internal ROM size is 8192 (8k) bytes and is memory mapped from $E000 to $FFFF.

3.6.2. R96V24AC Support Firmware

Firmware was added in the C40 internal masked ROM to support the 4 MHz RC96V24AC applications. Without the use of this internal firmware, software packages executing a 16-bit I/O write operation to load the upper and lower divisor latches of the 16550 interface require 10 MHz operation in order to perform two sequential writes to the interface with sufficient time between writes to ensure correct latching of the data by the MCU. Operation at 10 MHz in turn requires the use of faster (70 ns), more expensive EPROMs.

To execute from internal ROM, the TSTP pin should be a "no connect" which allows an internal resistor to pull the signal high. Typical applications have an external pull down resistor which should simply remove the resister to use the C40 in this application. The masked ROM code selects the slowest external access clock rate so that lower cost, slower (150 ns) EPROMs may be used. To take advantage of this feature, modified application firmware must be used which sets external accesses at one-half the clock rate (nominally 5 MHz) while taking into account that the MCU internal timers are running at 10 MHz.

3.7. INTERNAL RAM

The internal Random Access Memory (RAM) contains the user program stack and is used for scratch pad memory during system operation. This RAM is completely static in operation and requires no clock or dynamic refresh. The data contained in RAM is read out nondestructively with the same polarity as the input data. In the event that execution stops, RAM data is retained until execution resumes.

3.7.1. C40 Size and Location

The C40 internal RAM size is 1024 bytes. A block of 192 bytes is assigned to page 0 ($40 to $FF). Three blocks of 256 bytes each are assigned to pages 1 through 3 ($100 to $1FF, $200 to $2FF, and $300 to $3FF). A fifth block of 64 bytes is assigned to page 4 ($400 to $43F).

3.7.2. L39 Size and Location

The L39 internal RAM size is 1470 bytes. A block of 192 bytes is assigned to page 0 ($40 to $FF). Four blocks of 256 bytes each are assigned to pages 1 through 4 ($100 to $1FF, $200 to $2FF, $300 to $3FF, and $400 to $4FF). A sixth block of 254 bytes is assigned to page 5 ($500 to $5FD).

3.8. MEMORY BANKING

The MCU has a special dual port RAM that allows the user to bank select external memory (Figure 3-7). The dual port bank select RAM consists of 8 bytes of internal RAM shown below from address locations $0018 through $001F. These bytes are called Bank Select Registers 0-7 (BSR0-BSR7). Each Bank Select Register contains the following bits:

7	6	5	4	3	2	1	0
ES3	ES2	ES1	ES0	A16	A15	A14	A13

Each time the CPU provides addresses AD15, AD14, and AD13, one of the eight BANK select RAM is selected.

Address	CPU Address Line			Bank Select	Reset	Default	
Map	AD15	AD14	AD13	Register	Values	Select	Speed
$0018	0	0	0	0	1110 0000	ES0	X1
$0019	0	0	1	1	1101 0001	ES1	X1
$001A	0	1	0	2	1011 0010	ES2	X1
$001B	0	1	1	3	1011 0011	ES2	X1
$001C	1	0	0	4	0111 0100	ES3	X1
$001D	1	0	1	5	0111 0101	ES3	X1
$001E	1	1	0	6	0111 0110	ES3	X1
$001F	1	1	1	7	0111 0111	ES3	X1

The MCU outputs active low chip select lines on PB5, PB4, PB3, and PB2 according to the logic-0 bits of the selected BSR bits 4 through 7 (ES3, ES2, ES1, and ES0), respectively, when its respective port B select registers PBS5 = 0, PBS4 = 0, PBS3 = 0, or PBS2 = 0. PB1 outputs active low chip select ES4 for address space $0600 to $0800 when PBS1 = 0. The address space for ES4 is included when bank select RAM 0 is selected and PBS1 = 1.

The MCU outputs address lines A15, A14, and A13 according to the contents of the selected BSR bits 2-0, respectively. PB0 becomes bit 3 (A16) of the contents of the selected BSR when the port B select register PBS0 = 0. Address A16 allows addressing 8k bytes memory banks above 64k bytes of memory.

The user can alter the contents of each BSR byte by writing to the individual register address or use its respective reset value shown above.

The selected BSR's speed choice for chip selects ES3, ES2, ES1, and ES0 are reflected in the ESS speed register shown below. During power up, bank select 7 is automatically selected with chip select ES3 running at the highest X1 speed. The user can alter the contents of the ESS register or use the highest X1 speed default value shown.

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0033	ES Speed (ESS)	ES3		ES2		ES1		ES0	
		ESS7	ESS6	ESS5	ESS4	ESS3	ESS2	ESS1	ESS0
		MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB

Default = $00 (X1 - 1 clock time selected.)

The following table shows the memory speed selected for the chip selects chosen in the selected banking RAM.

ES Speed Control		
MSB	LSB	Speed
0	0	X 1
0	1	X 1/2
1	0	X 1/3
1	1	X 1/4

Priority logic makes the memory speed take on the choice of the highest active chip select ES3, ES2, or ES1 when two or more are enabled.

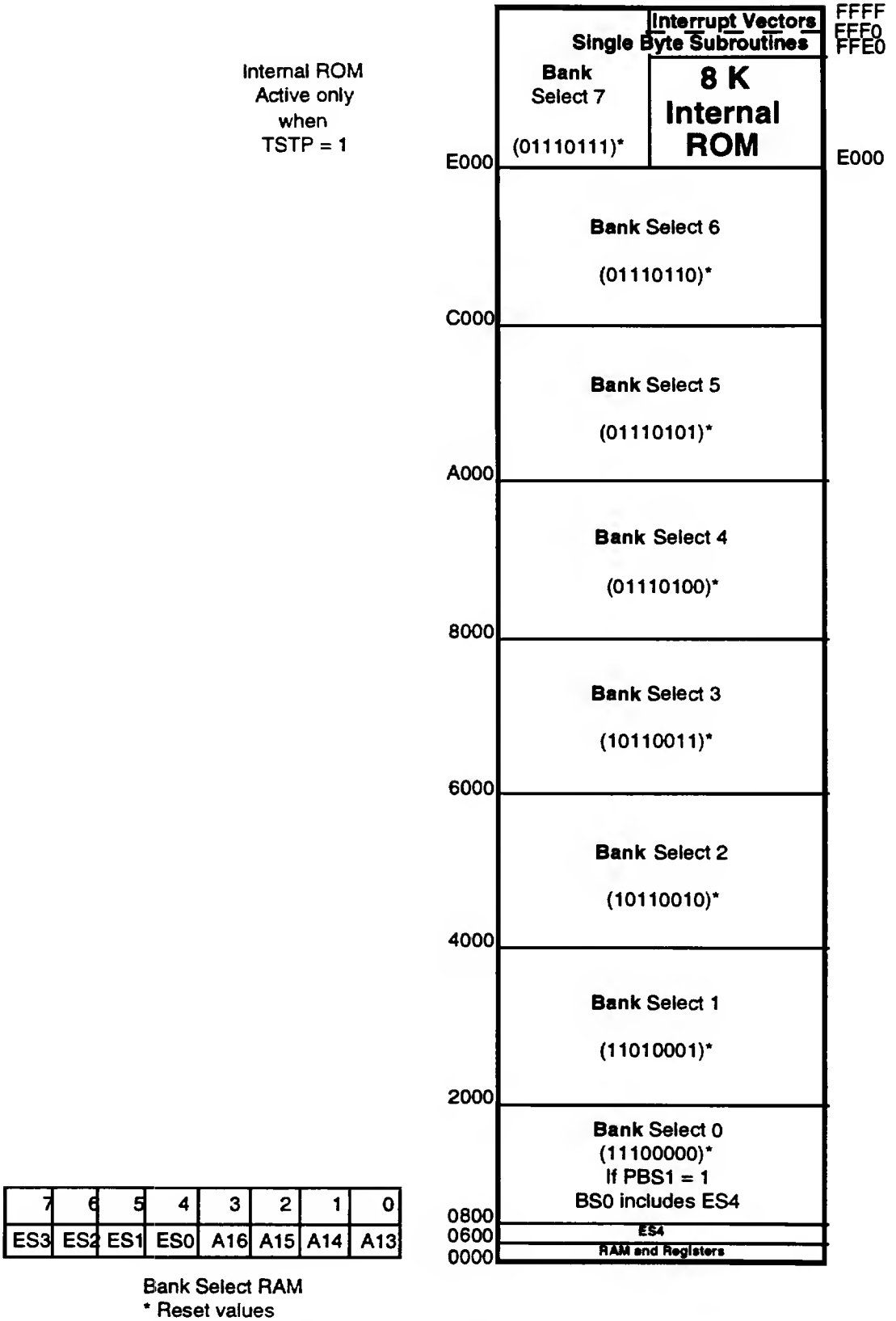
Memory banking is performed using internal address translation using eight Bank Select Registers (3.8.1). Each Bank Select Register (BSR) controls the address translation of A[15:13], controls the A16 pseudo-address bit, and defines an active bank select from ES[3:0] for a fixed 8k-byte region of linear address. This logic enables flexible banking while simultaneously eliminating the need for the external programmable logic device normally required to support banking. In addition, stretched internal clock cycles can be assigned independently to each BSR chip select range using the ES Speed Register (3.8.2) to allow the use of slower memories. The address translation allows several banks with the same logical address range to be physically relocated so that they can all reside within the same memory device with different physical address ranges.

This banking technique becomes more easily understood if one views the addition of A16 and ES[3:0] separately from the internal address translation. Although A16 functions identically to ES[3:0] in that it is a manually controlled bit in each BSR and drives an external pin, emulation support mandates that this signal not be used as a chip select. The new expanded address field A[16:0] maps directly to A[16:0] of a 128K memory device. When each ES signal drives the chip enable of a separate 128K memory device, this scheme supports a theoretical banked memory of 128K x 4 = 512K bytes.

The address translation feature does not increase the amount of banked memory beyond this theoretical limit, but it allows several banks with the same logical address range to be mapped within the same physical device.

Note that ES0-ES3 and A16 are Port B special purpose outputs and must be enabled using the Port B Select (PBS) register.

The memory banks are illustrated within the overall memory map in Figure 3-7. The MCU banking logic is shown in Figure 3-8. The bank select registers are shown in Table 3-6.



7	6	5	4	3	2	1	0
ES3	ES2	ES1	ES0	A16	A15	A14	A13

Bank Select RAM
* Reset values

Figure 3-7. Memory Map - RAM Banking

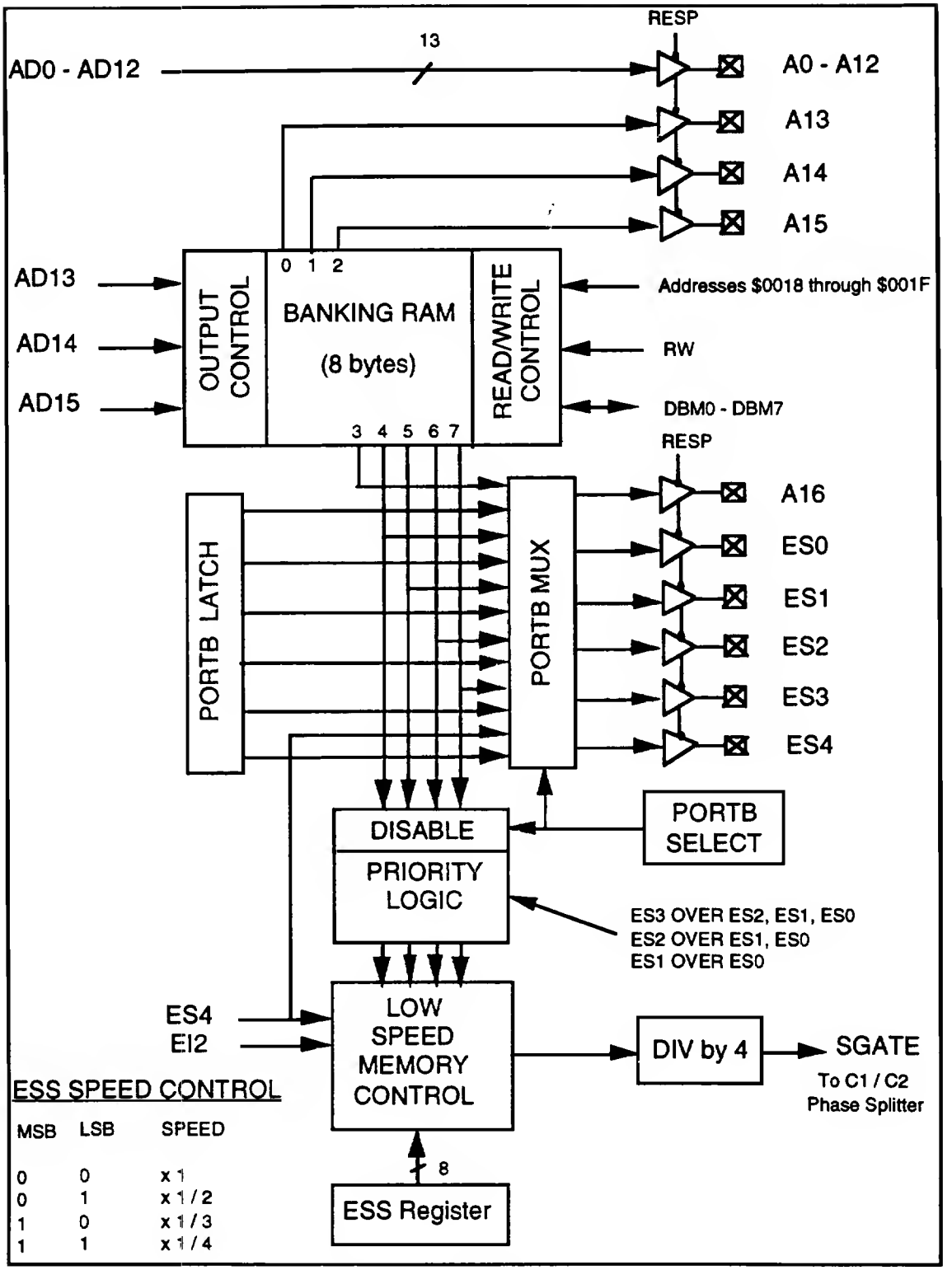


Figure 3-8. Banking Logic

Table 3-6. Register Bit Assignments: Bank Select Register I (BSRI) - 0018h-001Fh

Addr.	Function	Bit 7	6	5	4	3	2	1	0
0018	Bank Select Register 0 (BSR0)	ES3 (0)	ES2 (0)	ES1 (0)	ES0 (0)	A16 (0)	A15 (0)	A14 (0)	A13 (0)
0019	Bank Select Register 1 (BSR1)	ES3 (1)	ES2 (1)	ES1 (1)	ES0 (1)	A16 (1)	A15 (1)	A14 (1)	A13 (1)
001A	Bank Select Register 2 (BSR2)	ES3 (2)	ES2 (2)	ES1 (2)	ES0 (2)	A16 (2)	A15 (2)	A14 (2)	A13 (2)
001B	Bank Select Register 3 (BSR3)	ES3 (3)	ES2 (3)	ES1 (3)	ES0 (3)	A16 (3)	A15 (3)	A14 (3)	A13 (3)
001C	Bank Select Register 4 (BSR4)	ES3 (4)	ES2 (4)	ES1 (4)	ES0 (4)	A16 (4)	A15 (4)	A14 (4)	A13 (4)
001D	Bank Select Register 5 (BSR5)	ES3 (5)	ES2 (5)	ES1 (5)	ES0 (5)	A16 (5)	A15 (5)	A14 (5)	A13 (5)
001E	Bank Select Register 6 (BSR6)	ES3 (6)	ES2 (6)	ES1 (6)	ES0 (6)	A16 (6)	A15 (6)	A14 (6)	A13 (6)
001F	Bank Select Register 7 (BSR7)	ES3 (7)	ES2 (7)	ES1 (7)	ES0 (7)	A16 (7)	A15 (7)	A14 (7)	A13 (7)

3.8.1. Bank Select Register (BSR)

Bits 0-3: A13 (i) - A16(i). Address Translation Bits for Bank I.

Bits 4-7: ES0(i) - ES3(I). Memory Bank Select for Bank i.

Notes:

1. When 0 is not present in BSR7 (ES3 to ES0), the internal ROM is enabled. This allows a bootstrap routine in internal ROM to be run, e.g., to configure the ES Speed register to select the effective clock width during external memory access. When the internal ROM is enabled, the external transceivers are disabled for addresses within the BSR7 range.

   When the TSTP pin is high, the internal ROM is enabled following reset until a new value is written into BSR7.

3.8.2. ES Speed Register

The ES Speed register at address $0033 (Table 3-7) selects the number clock times that the internal clock is stretched during a memory access in an address range corresponding to an ES0-ES3 chip select.

Table 3-7. Register Bit Assignments - ES Speed Register - 0033h

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0033	ES Speed (ESS)	ES3-1	ES3-0	ES2-1	ES2-0	ES1-1	ES1-0	ES0-1	ES0-0

Default = $00 (X1 - 1 clock time selected.)

Bits 0-1: **Chip Select 0 (ES0) Effective Clock Width.** These two bits select the number of clock times that the internal clock is stretched during a memory access when the ES0 chip select is active.

Bit 1 (ES0-1)	Bit 0 (ES0-0)	Effective Clock Width
0	0	X1 - 1 clock time
0	1	X2 - 2 clock times
1	0	X3 - 3 clock times
1	1	X4 - 4 clock times

Bits 2-3: **Chip Select 1 (ES1) Effective Clock Width.** These two bits select the number of clock times that the internal clock is stretched during a memory access when the ES1 chip select is active.

Bit 3 (ES1-1)	Bit 2 (ES1-0)	Effective Clock Width
0	0	X1 - 1 clock time
0	1	X2 - 2 clock times
1	0	X3 - 3 clock times
1	1	X4 - 4 clock times

Bits 4-5: **Chip Select 2 (ES2) Effective Clock Width.** These two bits select the number of clock times that the internal clock is stretched during a memory access when the ES2 chip select is active.

Bit 5 (ES2-1)	Bit 4 (ES2-0)	Effective Clock Width
0	0	X1 - 1 clock time
0	1	X2 - 2 clock times
1	0	X3 - 3 clock times
1	1	X4 - 4 clock times

Bits 6-7: **Chip Select 3 (ES3) Effective Clock Width.** These two bits select the number of clock times that the internal clock is stretched during a memory access when the ES3 chip select is active.

Bit 7 (ES3-1)	Bit 6 (ES3-0)	Effective Clock Width
0	0	X1 - 1 clock time
0	1	X2 - 2 clock times
1	0	X3 - 3 clock times
1	1	X4 - 4 clock times

3.9. PARALLEL INPUT/OUTPUT PORTS

The MCU parallel input/output interface consists of five 8-bit ports: A, B, C, D, and E. Port E is available only on 80-pin PQFP and 84-pin PLCC packages.

Ports A, C and E contain 24 bidirectional lines with the data direction determined by the direction registers. Port D has 4 bidirectional lines (0-3) with the data direction determined by its direction register, and 4 input only lines (4-7). Port B supports 8 output-only lines.

All port lines can be used for general purpose functions. Thirty two I/O lines can be assigned special functions under software control. The special purpose and mask option functions of the port lines are identified in Table 3-8. Table 3-9 further defines the special purpose applications along with software control and direction register requirements. Port read and write timing is described in Section 5.

Output Mode. The data written to each output pin is loaded into an output data latch. The data will remain in the output latch until new data is written to the port address or until power is removed. The output latches are individually connected to output drivers. The output drivers are double-ended, push-pull type. The drivers force the output pins high (≥ 2.4V) if the output data bit is a logic 1, or low (≤0.4V) if the output data bit is a logic 0. The output drivers are TTL compatible.

Input Mode. For each input port line, either permanently or direction register assigned as an input, the data is sampled by an input synchronizer. A low input level (≤0.8 v) is interpreted as a logic 0 and a high input level (≥2.0 V) is interpreted as a logic 1. Input data is sampled during internal C2 clock time and then temporarily held from C1 to C1 clock time. When the CPU reads the input port, the data is transferred to the CPU during C2 time and represents data sampled during the previous C2 time.

Table 3-8. I/O Port Special Purpose Functions

Port Bit	Port				
	A	B	C	D	E
0	TIMA, RING, or CTSP	A16	HD0	HA0	None
1	TXDE or - Edge	ES4	HD1	HA1	None
2	TXD, LPWU TXD	ES3	HD2	HA2	None
3	TXCLK, +Edge, or HTACKP	ES2	HD3	HA3	None
4	TXREF, ± Edge, RLSDP, or HRACKP	ES1	HD4	HCSP, LPWU DTR, or LPWU HCSP	None
5	REXD or TXRDY	ES0	HD5	HWTP, LPWU AL, or LPWU HWTP	None
6	RXD or RXRDY	HDIS*	HD6	HRDP	None
7	RXCLK or - Edge	HINT	HD7	±Edge	None

* Not recommended for use.

Table 3-9. I/O Port Special Purpose Function Control

Port Line	Function	Option Selected by	I/O	Direction Reg.
Port A				
PA0	TIMA - Timer A Pulse Generator Output	TAM1 = 0, TAM0 = 1	O	PAD0 = 1
	TIMA - Timer A Event Counter Input	TAM1 = 1, TAM0 = 0	I	PAD0 = 0
	TIMA - Timer A Pulse Width Measurement Input	TAM1 = 1, TAM0 = 1	I	PAD0 = 0
	RING - PA0 High (Ring)	LPR5 = 1	I	PAD0 = 0
	CTSP - USART TXD Sync to Falling Edge	SMR6 = 1, SMR1 = 1	I	PAD0 = 0
PA1	TXDE - USART PA1 Output Copies PA2 Input	SF7 = 1	O	PAD1 = 1
	PA1 - Negative Edge Detect	None, see EI7	I/O	PAD1 = X
PA2	TXD - USART Serial Input	SMR6 = 1	I	PAD2 = 0
	LPWU - PA2 Low (TXD)	LPR3 = 1	I	PAD2 = 0
PA3	TXCLK - USART Internal Timing		O	PAD3 = 1
	TXCLK - USART External Timing		I	PAD3 = 0
	TXCLK - USART Copy TXCLK (PA4)	SMR4 = 1	I/O	PAD3 = X
	PA3 - Positive Edge Detect	None, see SIR3	I/O	PAD3 = X
	HTACKP - TX Acknowledge Input	HCR2 = 1, HCR1 = 1, FCR) = 1, SMR7 = 0, SMR4 = 0	I	PAD3 = 0
PA4	TXREF - External Clock Input	SMR4 = 1 and SMR2 = 1		PAD4 = 0
	PA4 - Positive and Negative Edge Detect	None, see IER6	I/O	PAD4 = X
	RLSDP USART RXD Sync to Falling Edge	SMR7 = 1, SMR1 = 1		PAD4 = 0
	HRACKP - 16550A RX Acknowledge Input	HCR2 = 1, HCR1 = 1, FCR) = 1, SMR7 = 0, SMR4 = 0	I	PAD4 = 0
PA5	REXD - USART Serial Input	SM7 = 1, SF5 = 0, SF6 = 1		PAD5 = 0
	TXRDY - TX Ready Output	HCR2 = 1, HCR1 = 1, FCR) = 1, SMR7 = 0, SMR4 = 0		PAD5 = 1
PA6	RXD - USART Serial Output	SMR7 = 1		PAD6 = 0
	RXRDY - RX Ready Output	HCR2 = 1, HCR1 = 1, FCR) = 1, SMR7 = 0, SMR4 = 0		PAD6 = 1
PA7	RXCLK - RXD Serial Output Clock	SMR7 = 1, SMR4 = 1	O	PAD7 = 1
	PA7 - Falling Edge Detect	None, see SIR5	I	PAD7 = X
Port B				
PB0	A16 - External Bus Address Line A16	PBS0 = 0	O	Output Only
PB1	ES4 - External Bus Chip Select	PBS1 = 0	O	
PB2	ES3- External Bus Chip Select	PBS2 = 0	O	
PB3	ES2- External Bus Chip Select	PBS3 = 0	O	
PB4	ES1- External Bus Chip Select	PBS4 = 0	O	
PB5	ES0- External Bus Chip Select	PBS5 = 0	O	
PB6	HDIS - Host Bus Driver Disable*	PBS6 = 0, HCR2 = 1	O	
PB7	HINT - Host Bus Interrupt Line	PBS7 = 0, HCR2 = 1	O	
* Not recommended for use.				

Table 3-9. I/O Port Special Purpose Function Control (Cont'd)

Port Line	Function	Option Selected by	I/O	Direction Reg.
		Port C		
PC0	HD0 - Host Bus Data Line 0	HCR2 = 1	I/O	Controlled by
PC1	HD1 - Host Bus Data Line 1			HWTP and
PC2	HD2- Host Bus Data Line 2			HRDP
PC3	HD3- Host Bus Data Line 3			
PC4	HD4- Host Bus Data Line 4			
PC5	HD5- Host Bus Data Line 5			
PC6	HD6- Host Bus Data Line 6			
PC7	HD7- Host Bus Data Line 7			
		Port D		
PD0	HA0 - Host Bus Address Line 0	HCR2 = 1	I	N/A
PD1	HA1- Host Bus Address Line 1	HCR2 = 1		
PD2	HA2- Host Bus Address Line 2	HCR2 = 1		
PD3	HA3- Host Bus Address Line 3	HCR2 = 1 and HCR1 =0		
PD4	HCSP- Host Bus Chip Select	HCR2 = 1	I	input Only
	LPWU - PD4 Low (DTR)	LPR2 = 1		
	LPWU - PD4 (HCSP) Low and PD5 Low (HWTP)	LPR4 = 1		
PD5	HWTP - Host Bus Write Enable	HCR2 = 1		
	LPWU - PD5 Low (AL)	LPR1 = 1		
	LPWU - PD4 (HCSP) Low and PD5 Low (HWTP)	LPR4 = 1		
PD6	HRDP - Host Bus Read Enable	HCR2 = 1		
PD7	PA7 Falling or rising edge detect	None, see IER5		

3.9.1. Bidirectional Ports A, C, and E

Ports A, C, and E consist of 24 general purpose bidirectional input/output lines. The data direction for each I/O line is controlled by an associated direction register bit. For each direction register bit that is a logic 1, the corresponding port line is an output. Conversely, a 0 in a direction register bit defines the corresponding port line as an input. The direction register bits are initialized to a 0 by reset causing the I/O ports to be inputs.

All port A, and C lines can be assigned to special purpose functions during operation under software control. The port A lines can be assigned to special functions under software control (Tables 3-8 and 3-9). PA3 and PA7 have associated edge detect logic that can generate an IRQ interrupt.

Seven port B lines can be assigned to special functions under software control and one line can be permanently masked to a special function (Tables 3-8 and 3-9). PB2 and PB3 have associated edge detect logic that can generate an IRQ interrupt.

All eight port C lines can be assigned to host bus data lines under software control (Tables 3-8 and 3-9).

3.9.2. Bidirectional and Input Only Port D

The eight port D lines are grouped as four general purpose bidirectional input/output lines and as four input only lines (Tables 3-8 and 3-9). PD7 has associated edge detect logic that can generate an IRQ interrupt. Seven lines (PD0-PD6) can be assigned by software as host bus address and control line inputs (3.11). The direction register control bits for port D must be set high (1) for output and low (0) for input. The direction register control bits are initialized to 0 by reset.

3.9.3. Output Port B

The 8 port B lines (PB0-PB7) are general or special purpose output only (Table 3-8). Lines PB0-PB5 can be assigned to A16 address line and the ES0-ES4 chip select functions controlled by the Bus Select Registers and PB6 and PB7 can be assigned to HDIS and HINT functions (see Table 3- 9), however, it is recommended that the host bus I/O read and chip select inputs to the MCU be used as the host bus driver disable function instead of HDIS to control data direction and enable external transceiver data output onto the host bus. Port B output latches are initialized low upon reset. The Port B output drivers tri-state (float) during reset active low.

3.9.4. Port B Select Register (PBS)

The Port B Select Register (PBS) bits are identified in Table 3-10.

Table 3-10. Port B Select Register (PBS)

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0005	Port B Select (0-7)	PBS7 0-- HINT 1-- PB7	PBS6 0-- HDIS 1-- PB6	PBS5 0-- ES0 1-- PB5	PBS4 0-- ES1 1-- PB4	PBS3 0-- ES2 1-- PB3	PBS2 0-- ES3 1-- PB2	PBS1 0-- ES4 1-- PB1	PBS0 0-- A16 1-- PB0

Bit 0: **PB0 /A16 Select. Control bit.** Control bit.

0 = Selects PB0 to operate as A16.

1 = Selects PB0 to operate as a general purpose port.

Bit 1: **PB1 /ES4 Select.** Control bit.

0 = Selects PB1 to operate as ES4.

1 = Selects PB1 to operate as a general purpose port.

Bit 2: **PB2 /ES3 Select.** Control bit.

0 = Selects PB2 to operate as ES3.

1 = Selects PB2 to operate as a general purpose port.

Bit 3: **PB3/ES2 Select.** Control bit.

0 = Selects PB3 to operate as ES2.

1 = Selects PB3 to operate as a general purpose port.

Bit 4: **PB4/ES1 Select.** Control bit.

0 = Selects PB4 to operate as ES1.

1 = Selects PB4 to operate as a general purpose port.

Bit 5: **PB5/ES0 Select.** Control bit.

0 = Selects PB5 to operate as ES0.

1 = Selects PB5 to operate as a general purpose port.

Bit 6: **PB6 /HDIS Select.** Control bit.

0 = Selects PB6 to operate as HDIS (not recommended for use).

1 = Selects PB6 to operate as a general purpose port.

Bit 7: **PB7/HINT Select.** Control bit.

0 = Selects PB7 to operate as HINT.

1 = Selects PB7 to operate as a general purpose port.

3.9.5. External Interrupt Register (EIR)

The External Interrupt Register (EIR) enables and reports interrupts associated with ports PA1, PA4, and PD7 (Table 3-11). All bits are cleared to zero by reset. The CPU can set or reset bits 0-4 by writing to address $000A. Bits 5-7 are cleared by writing a 0 to the corresponding bit position in the CIR. The CPU can monitor all bits by reading address $000A.

Table 3-11. External Interrupt Register (EIR) - $000A

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
000A	External Interrupt Register (EIR)	External Interrupt Flag PA1	External Interrupt Flag PA4	External Interrupt Flag PD7	External Interrupt Enable PA1 (IRQ5)	External Interrupt Enable PA4 (IRQ2)	External Interrupt Enable PD7 (IRQ1)	Edge Detect Polarity PA4 0=↓ Edge 1=↑ Edge	Edge Detect Polarity PD7 0=↓ Edge 1=↑ Edge

Bit 0: **PD7 Positive Edge Detect.** Control bit.

1 = Enables positive or rising (low-to-high) edge detection on PD7.

0 = Enables negative or falling (high-to-low) edge detection on PD7.

Bit 1: **PA4 Positive Edge Detect.** Control bit.

1 = Enables positive or rising (low-to-high) edge detection on PA4.

0 = Enables negative or falling (high-to-low) edge detection on PA4.

Bit 2: **PD7 Interrupt Enable.** Control bit.

1 = Enables assertion of IRQ1 when EIR5 is set to a logic 1.

0 = Disables assertion of IRQ1 due to EIR5.

Bit 3: **PA4 Interrupt Enable.** Control bit.

1 = Enables assertion of IRQ2 when EIR6 is set to a logic 1.

0 = Disables assertion of IRQ2 due to EIR6.

Bit 4: **PA1 Interrupt Enable.** Control bit.

1 = Enables assertion of IRQ5 when EIR7 is set to a logic 1.

0 = Disables assertion of IRQ5 due to EIR7.

Bit 5: **PD7 Interrupt Flag.** Status bit.

1 = A positive (EIR0 = 1) or negative (EIR0 = 0) edge has been detected on PD7. This bit is cleared by writing a logic 0 to CIR5.

0 = An edge has not been detected on PD7.

Bit 6: **PA4 Interrupt Flag.** Status bit.

1 = A positive (EIR1 = 1) or negative (EIR1 = 0) edge has been detected on PA4. This bit is cleared by writing a logic 0 to CIR6.

0 = An edge has not been detected on PA4.

Bit 7: **PA1 Interrupt Flag.** Status bit.

1 = A negative edge has been detected on PA1 This bit is cleared by writing a logic 0 to CIR7.

0 = An edge has not been detected on PA1.

3.9.6. Clear Interrupt Register (CIR)

The Clear Interrupt Register (CIR) at address $000B (Table 3-12) is used to clear five interrupt flags (PA3, PA7, PD7, PA4, and PA1) and set the fast/slow operation of ES4 expansion bus addressing. CIR0, CIR1 and CIR2 are initialized to 0 by reset.

Table 3-12. Clear Interrupt Register (CIR)- $000B

Addr.	Function	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
000B	Clear Interrupt Register (CIR)	Clear interrupt Flag PA1 (EIR7)	Clear interrupt Flag PA4 (EIR6)	Clear Interrupt Flag PD7 (EIR5)	Clear interrupt Flag PA7 (SIR5)	Clear Interrupt Flag PA3 (SIR6)	ES4 Fast Memory Cycle	Not Used	Not Used

Bits 0-1: Not Used.

Bit 2: ES4 Expansion Bus Fast Memory Cycle. Control and status bit. This bit can be set or reset by the CPU. The CPU can monitor the bit by reading address $000B. The bit is initialized to zero by reset.

1 = Enables ES4 Expansion Bus addresses to operate at fast speed, i.e., with one ø2 clock cycle per memory access cycle.

0 = Enables ES4 Expansion Bus addresses to operate at slow speed, i.e., with two ø2 clock cycles per memory access cycle.

Bit 3: Clear PA3 Interrupt Flag. Control bit. Reading this bit position always returns a 1.

1 = No effect.

0 = Resets the PA3 Interrupt Flag (SIR6) to a logic 0.

Bit 4: Clear PA7 Interrupt Flag. Control bit. Reading this bit position always returns a 1.

1 = No effect.

0 = Resets the PA7 Interrupt Flag (SIR5) to a logic 0.

Bit 5: Clear PD7 Interrupt Flag. Control bit. Reading this bit position always returns a 1.

1 = No effect.

0 = Resets the PD7 Interrupt Flag (EIR5) to a logic 0.

Bit 6: Clear PA4 Interrupt Flag. Control bit. Reading this bit position always returns a 1.

1 = No effect.

0 = Resets the PA4 Interrupt Flag (EIR6) to a logic 0.

Bit 7: **Clear PA1 Interrupt Flag.** Writing a logic 0 to this bit position resets the PA1 Interrupt Flag (EIR7) to a logic 0. Writing a logic 1 to this bit position has no effect. Reading this bit position always returns a 1.

1 = No effect.

0 = Resets the PA1 Interrupt Flag (EIR7) to a logic 0.

3.10. COUNTER/TIMERS

There are two separate 16-bit counter/timer systems in the MCU: Counter/Timer A (called Timer A) and Counter/Timer B (called Timer B). Timer A operates in one of four modes and can drive an output port. Timer B operates only in the interval timer mode with no output port option. Otherwise, operation of the two counter/timers is similar except for register addresses, the generated IRQ (and priority level) and the interfacing I/O port. The operation of Timer A is described in detail followed by a description of Timer B differences. Block diagrams of Timer A and Timer B are shown in Figures 3-9 and 3-10, respectively.

A divide-by-32 counter connected to Ø2 clock is shared by both timers. The counter provides a Ø2/32 clock that can be individually selected by each timer.

3.10.1. Timer A Registers

Timer A is composed of a 16-bit latch, a 16-bit counter and an 8-bit snapshot register (Figure 3-9). The latch consists of two 8-bit registers, Timer A Upper Latch (TAUL) and Timer A Lower Latch (TALL). The counter also consists of two 8-bit registers, Timer A Upper Counter (TAUC) and Timer A Lower counter (TALC). The snapshot register is referred as Timer A Snapshot (TAS). Timer A operation is controlled and monitored using the Timer A Mode Register (Table 3-13).

TALL is loaded by the CPU writing to address $0011. TAUL can be loaded by writing to either $0012 or $0013. When the CPU writes to address $0013, the contents of TALL and TAUL are also downloaded into TALC and TAUC, respectively, and the Timer A Interrupt Flag (TAM7) is cleared. The contents of TALC can be monitored at any time by reading $0011. Reading $0011 also causes the contents of TAUC to transfer into TAS. The contents of TAS can be monitored by reading either $0012 or $0013.

When Timer A underflows, the Timer A Interrupt Flag bit in the (TAM7) is set to a logic 1. This bit can be used to assert IRQ5.

3.10.2. Timer B Registers

Timer B is to Timer A except only the interval timer mode is supported (Figure 3-10). Timer B registers are located at $0014-$0017. When the Timer B Underflow Flag is set (TBM7) and enabled (TBM5 and TBM6), IRQ3 is asserted. Timer B interfaces with I/O port PB0 rather than PA0.

Timer B is composed of a 16-bit latch, a 16-bit counter and an 8-bit snapshot register. The latch consists of two 8-bit registers, Timer B Upper Latch (TBUL) and Timer B Lower Latch (TBLL). The counter also consists of two 8-bit registers, Timer B Upper Counter (TBUC) and Timer B Lower counter (TBLC). The snapshot register is referred as Timer B Snapshot (TBS).

TBLL is loaded by the CPU writing to address $0015. TBUL can be loaded by writing to either $0016 or $0017. When the CPU writes to address $0017, the contents of TBLL and TBUL are also downloaded into TBLC and TBUC, respectively, and the Timer B Interrupt Flag (TBM7) is cleared. The contents of TBLC can be monitored at any time by reading $0015. Reading $0015 also causes the contents of TBUC to transfer into TBS. The contents of TBS can be monitored by reading either $0016 or $0017.

When Timer B underflows, the Timer B Interrupt Flag bit in the (TBM7) is set to a logic 1. This bit can be used to assert IRQ3.

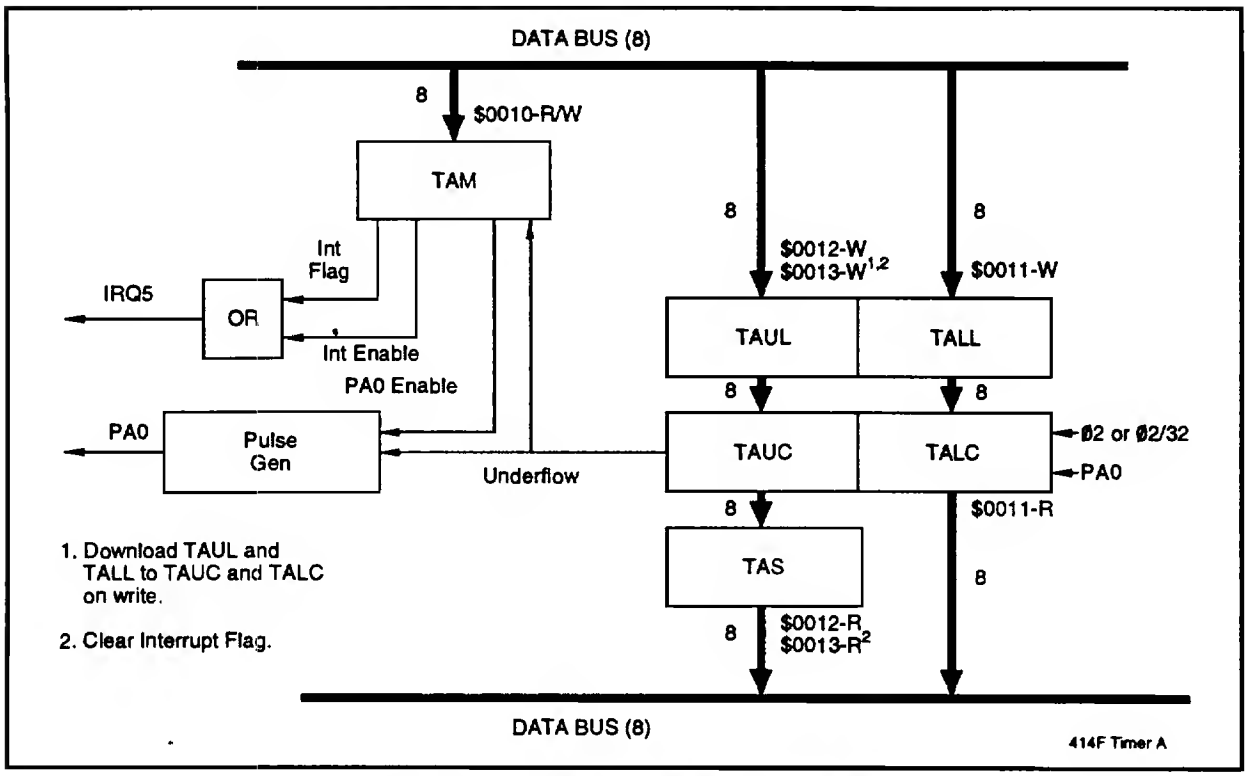


Figure 3-9. Counter/Timer A Block Diagram

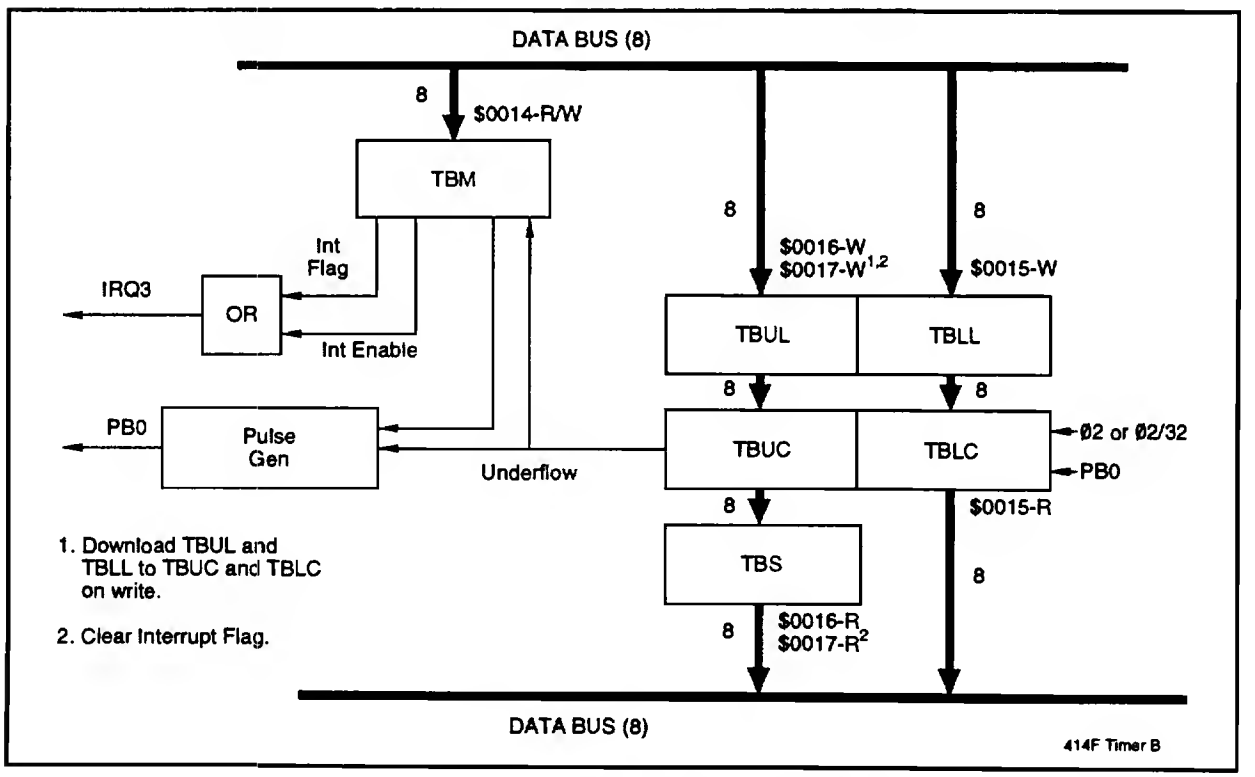


Figure 3-10. Counter/Timer B Block Diagram

3.10.3. Timer A Mode Register (TAM)

The Timer A Mode Register (TAM) selects the Timer A operating mode selection, and controls and reports the Timer A interrupt (Table 3-13). Bits 0-2, 5, and 6 are cleared to zero by reset, and can be reset or set by the CPU writing to address $0010. Bits 0-2, and 5-7 can be read by the CPU.

Table 3-13. Register Bit Assignments - 0010h - 0013h

Addr.	Function	Bit 7	6	5	4	3	2	1	0
0010	Timer A Mode (TAM)	Timer A Interrupt Flag	Timer A Interrupt Enable	Timer A Vector Select	Not Used		Timer A Div by 32 Prescale	Timer A Mode Bit 1 (TAM1)	Timer A Mode Bit 0 (TAM0)
0011	Timer A Lower Latch	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0012	Timer A Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0013	Timer A Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8

Bits 0-1: **Timer A Mode Select.** These two bits select the Timer A operating mode. The interval timer mode is selected upon reset since these bits are reset to logic 0.

Bit 1	Bit 0	Mode
0	0	Interval Timer
0	1	Pulse Generator
1	0	Event Counter
1	1	Pulse Width Measurement

Bits 2: **Timer A Divide by 32 Prescale** Control bit.

1 = The Timer A ø2 clock is divided by 32.

0 = The Timer A ø2 clock is not divided by 32.

Bits 3-4: **Not used.**

Bit 5: **Timer A IRQ5 RAM Vector Enable.** Control bit.

1 = When the Timer A Interrupt Enable (TAM6) is a logic 1, IRQ5 is asserted through the Timer A IRQ5 RAM vector.

0 = When the Timer A Interrupt Enable (TAM6) is a logic 1, IRQ5 is asserted through the Timer A IRQ5 ROM vector.

Bit 6: **Timer A Interrupt Enable.** Control bit.

1 = Enables IRQ5 to be asserted when the Timer A Interrupt Flag (TAM7) is set to a logic 1.

0 = Disables IRQ5 assertion due to TAM7.

Bit 7: **Timer A Interrupt Flag.** Status bit. Reading or writing $0013 clears the Timer A Interrupt Flag (TAM7).

1 = Timer A counter underflow has occurred, i.e., it decremented from 0 to - 1 (0000 to $FFFF).

0 = Timer A counter underflow has not occurred.

3.10.4. Timer B Mode Register (TBM)

The Timer B Mode Register (TBM) controls and reports the Timer B interrupt (Table 3-14). Bits 0-2, 5, and 6 are cleared to zero by reset, and can be reset or set by the CPU writing to address $0014. Bits 0-2, and 5-7 can be-read by the CPU.

Table 3-14. Register Bit Assignments - 0014h - 0017h

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0014	Timer B Mode (TBM)	Timer B Interrupt Flag	Timer B Interrupt Enable	Timer B Vector Select	Not Used		Timer B Div by 32 Prescale	0	0
0015	Timer B Latch Low	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0016	Timer B Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0017	Timer B Upper Latch	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8

Bits 0-1: Not Used.

Bits 2: Timer B Divide by 32 Prescale. Control bit.

1 = The Timer B ø2 clock is divided by 32.

0 = The Timer B ø2 clock is not divided by 32.

Bits 3-4: Not used.

Bit 5: Timer B IRQ3 RAM Vector Enable. Control bit.

1 = When the Timer B Interrupt Enable (TBM6) is a logic 1, IRQ3 is asserted through the Timer B IRQ3 RAM vector.

0 = When the Timer B Interrupt Enable (TBM6) is a logic 1, IRQ3 is asserted through the Timer B IRQ3 ROM vector.

Bit 6: Timer B Interrupt Enable. Control bit.

1 = Enables IRQ3 to be asserted when the Timer B Interrupt Flag (TBM7) is set to a logic 1.

0 = Disables IRQ3 assertion due to TBM7.

Bit 7: Timer B Interrupt Flag. Status bit. Reading $0017 clears the Timer B Interrupt Flag (TBM7).

1 = Timer B counter underflow has occurred, i.e., it decremented from 0 to - 1 (0000 to $FFFF).

0 = Timer B counter underflow has not occurred.

3.10.5. Timer Modes

Since mode operation of both counter/timers is similar, mode operation is described for Timer A with Timer B operation indicated in parentheses. The waveforms for the four timer modes are illustrated in Figures 3-11 through 3-14.

Mode 0 - Interval Timer

Writing to TAUL (TBUL) transfers the 16-bit latch value to the counter. The counter counts down at the Ø2 or Ø2/32 rate. When the counter counts through zero, the TAIF (TBIF) is set to a 1, the value in the latches is transferred to the counter and the counter continues to count down. (See Figure 3-11.)

Mode 1 - Pulse Generation

The PAD0 (PBD0) direction register bit must be set to a 1 to establish PA0 (PB0) as an output pin before starting this mode. Writing to TAUL (TBUL) forces the PA0 (PB0) output low and starts the timer. Each time the timer counts through zero, the PA0 (PB0) output changes state to generate a square wave at a rate dependent upon the value loaded into the latches. The timer counts at either the Ø2 or Ø2/32 rate. Each time the counter counts through zero, the latch values are automatically transferred to the timer registers and the TAIF (TBIF) is set to a 1. (See Figure 3-12.)

Mode 2 - Event Counter

The PAD0 (PBD0) direction register bit must be set to 0 to establish PA0 (PB0) as an input pin. The TAM2 (TBM2) clock divide-by-32 bit must be set to a 0 to select divide-by-1. The counter is initialized with the latch value when the TAUL (TBUL) value is written to address $0013 ($0017). The timer decrements by 1 at each positive transition on input port PA0 (PB0). TAIF (TBIF) is set to a 1 when the counter counts through zero. At the same time the latch value is reloaded into the counter. The maximum rate of the signal of PA0 (PB0) is one-half the timer clock rate. (See Figure 3-13.)

Mode 3 - Pulse Width Measurement

The PAD0 (PBD0) direction register bit must be set to an 0 to establish PA0 (PB0) as an input pin. Writing to TAUL (TBUL) at $0013 ($0017) transfers the 16-bit latch value to the counter. The value in the timer is decremented at the Ø2 or Ø2/32 rate when the PA0 (PB0) signal is low. Each time the PA0 (PB0) signal goes high, the counter stops and then continues when the signal is low again. If the counter counts through zero, TAIF (TBIF) is set to a 1 and the latch value transfers to reinitialize the counter. The countdown continues as long as PA0 (PB0) is low. (See Figure 3-14.)

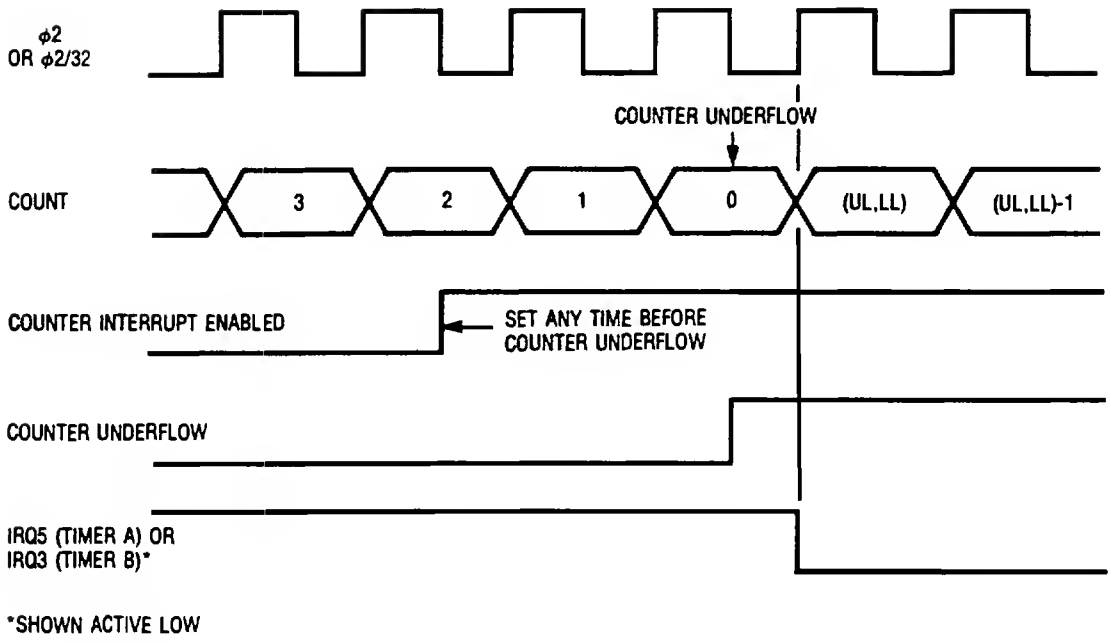


Figure 3-11. Interval Timer (Mode 0) Waveforms

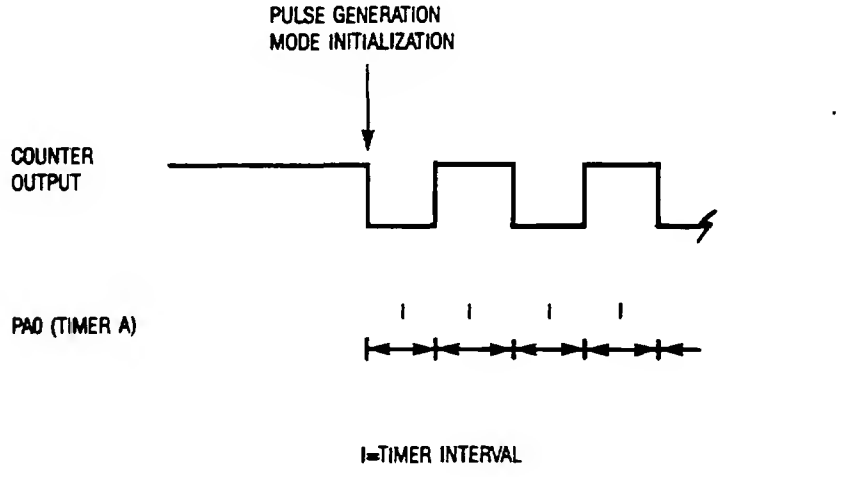


Figure 3-12. Pulse Generator (Mode 1) Waveforms

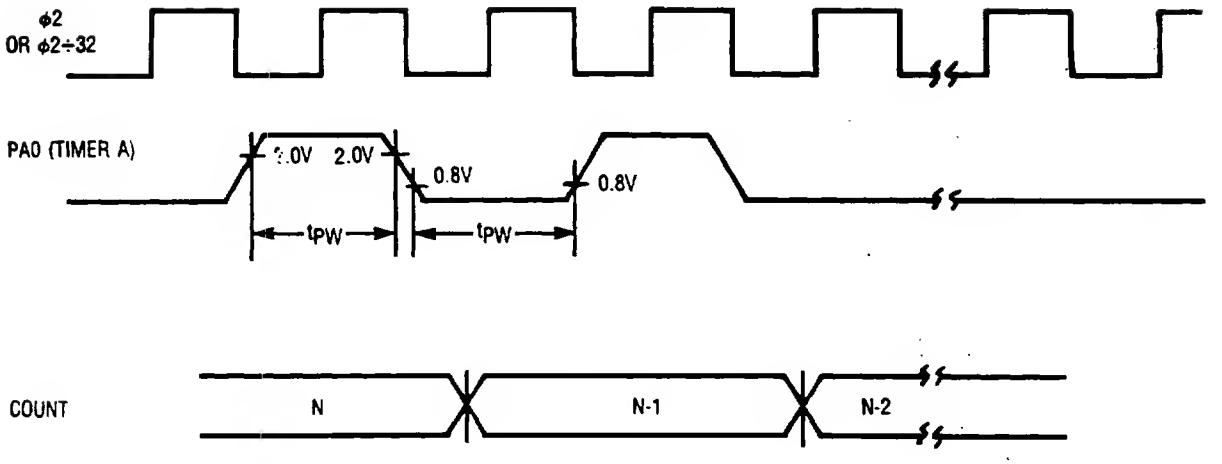


Figure 3-13. Event Counter (Mode 2) Waveforms

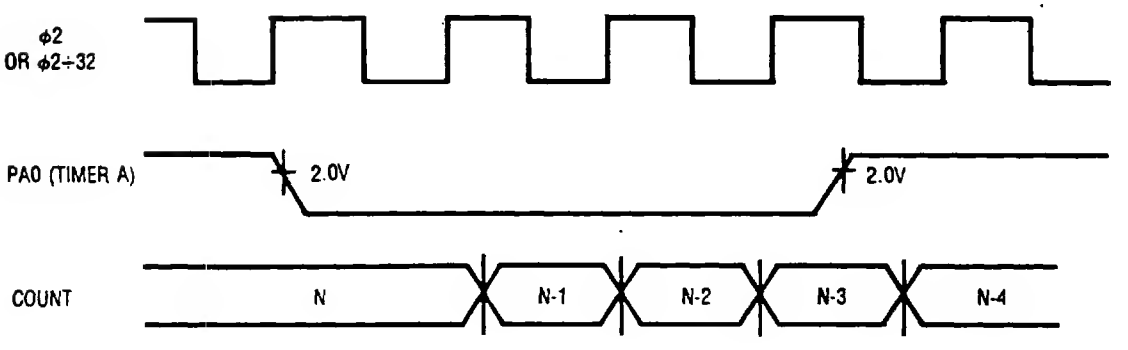


Figure 3-14. Pulse Width Measurement (Mode 3) Waveforms

3.11. PRECISION TIME GENERATORS

There are two identical 17-bit precision time generators: Precision Time Generator A (PTGA) and Precision Time Generator B (PTGB). Each PTG can be used for such functions as timing event interrupts, generating an external pulse train or as a source for synchronous USART timing.

Only PTGA is discussed in detail since both precision timer generators are identical in structure. Only the differences in I/O port addresses and IRQ interfaces are described. Block diagrams of PTGA and PTGB are shown in Figures 3-15 and 3-16, respectively.

3.11.1. Precision Time Generator A

PTGA consists of five 8-bit registers and a 17-bit pulse accumulator (Figure 3-15). The three input registers—PTGA Buffer (PAB), PTGA Lower Latch (PALL) and PTGA Upper Latch (PAUL)—are all 8-bit. There are two output registers: an 8-bit PTGA Lower Residue (PALR) and a 9-bit PTGA Upper Residue (PAUR). The PTGA Accumulator (PAAC)is 17-bits long. Operation is controlled by the PTGA Mode Register (PAM) located at $0034.

The CPU can read or write the contents of PAB using address $0035. The CPU can read or write the contents of PAUL at either address $0036 or $0037. Whenever the CPU writes to $0036 or $0037 the contents of PAB are transferred to PALL. This allows a simultaneous 16-bit update of the input latches. The 17th input bit to the pulse accumulator is always a logic 0. When the CPU writes to address $0037, PAB is transferred to PALL and the new contents of PALL and PAUL are downloaded into PALR and PAUR, respectively, and the most significant bit of PAUR is set to a logic 0. This feature is particularly helpful during testing.

Operation of the precision time generator is governed by the equation:

$$Rate = Latch^{*}(Ø2/2^{17})$$

where:

Rate = Pulse rate in Hz

Latch = Latch value

Ø2 = Internal clock rate in Hz

For example, if Ø2 = 4 MHz and the latch is loaded with 7550 ($107E), the resulting rate is 230408 Hz.

Conversely. the latch value can be computed using the equation

$$Latch = Rate^{*}(217/Ø2)$$

For example, if the desired rate = 342857 Hz, ø2 = 6 MHz, the required latch value is 7489.8 ($1D42).

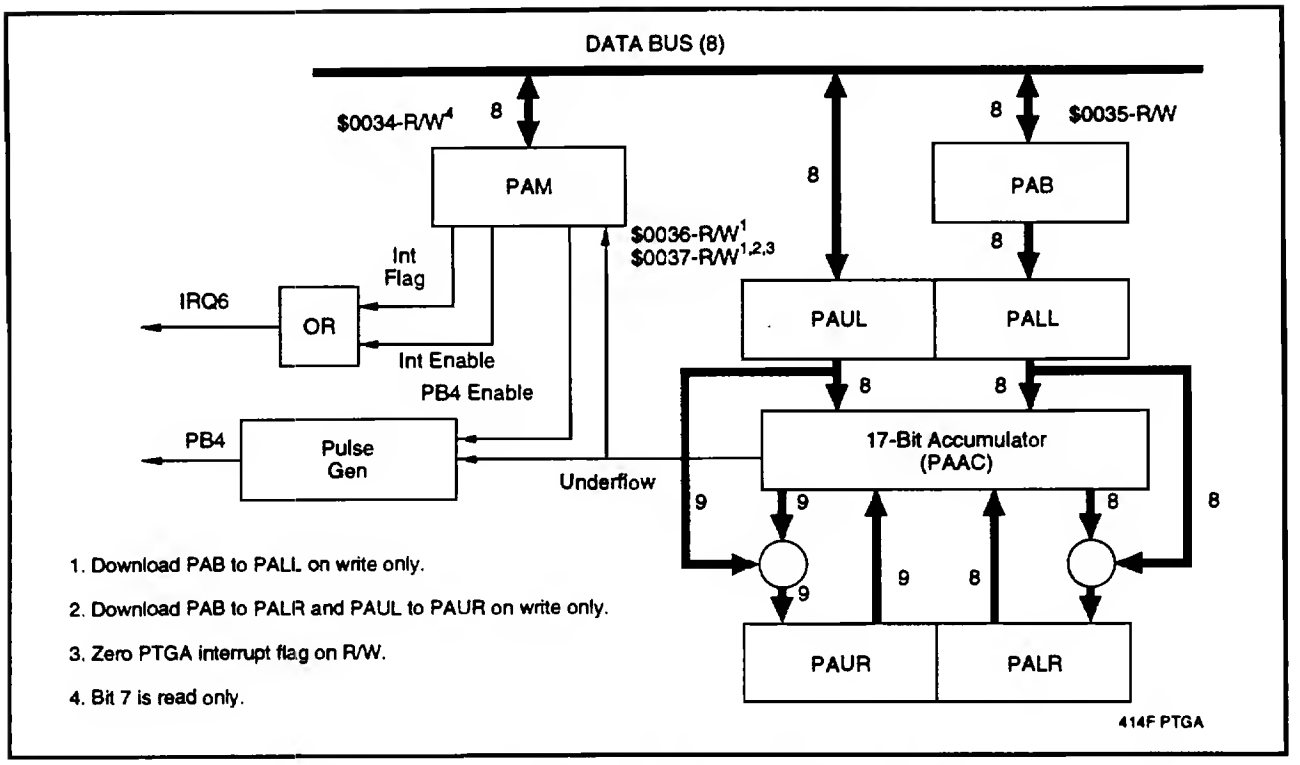


Figure 3-15. Precision Time Generator A Block Diagram

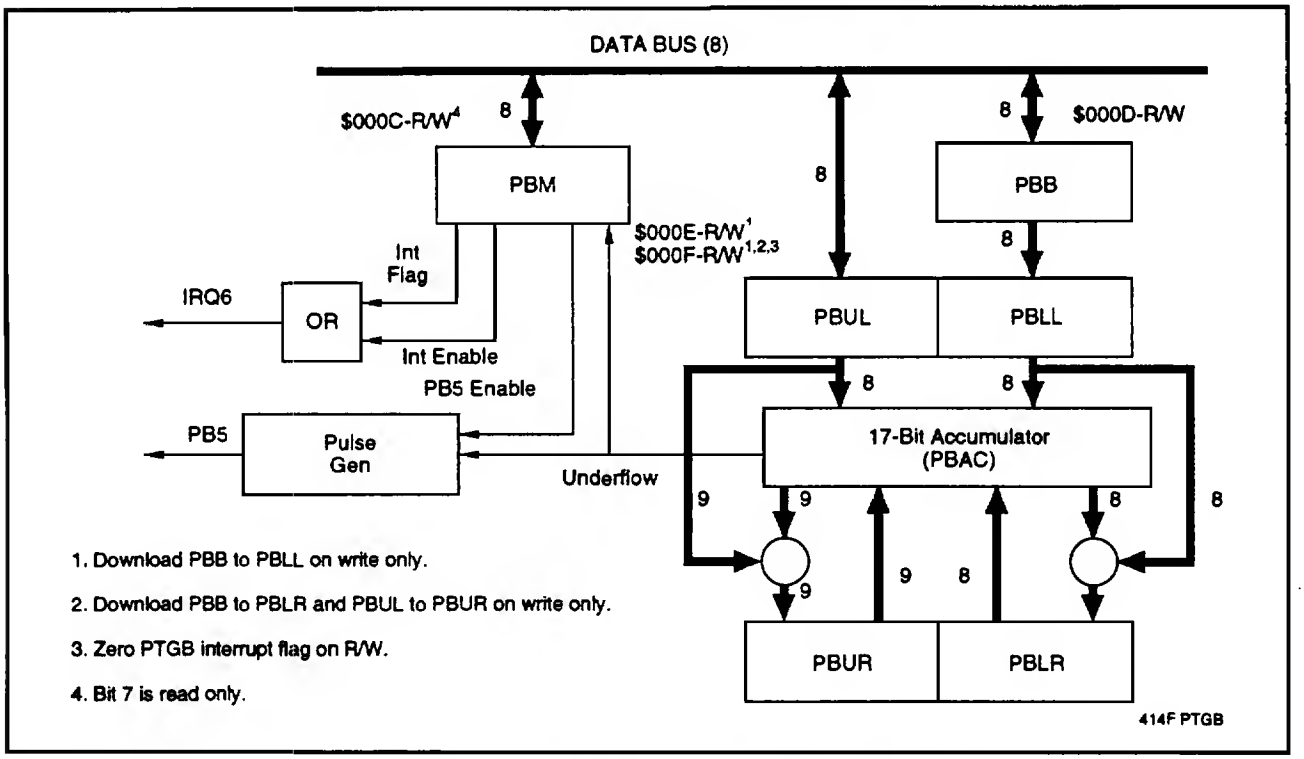


Figure 3-16. Precision Time Generator B Block Diagram

3.11.2. PTGA Mode Register (PAM)

The PTGA Mode Register (PAM) selects the PTGA timer and port options, and controls and reports the PTGA interrupt (Table 3-15). Bits 0, 1, and 6 are cleared by reset and can be set to a logic 1 or 0 by writing to address $0037. All bits can be read by the CPU.

Table 3-15. Register Bit Assignments - PTGA - 0034

Addr.	Function	Bit 7	6	5	4	3	2	1	0
0034	Precision Time Generator A (PTGA) Mode (PAM)	PTG A Interrupt Flag	PTG A Interrupt Enable	Not Used				TXD (SIN) PTG Select 1= PTGA 0 = PTGB	PTGA Timer Mode
0035	PTGA Buffer (PAB)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0036	PTGA Upper Latch (PAUL)	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
0037	PTGA Upper Latch (PAUL)	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8

Bit 0: **PTGA Timer Mode Select.** Control bit.

1 = PTGA operates as a timer.

0 = PTGA operates as a precision time generator.

Bit 1: **SIN (TXD) PTG Select.** Control bit.

1 = When SMR5 is a logic 1, selects PTGA as the input to the SINC Programmable Counter. The PTGA input may be divided by 3 (see SFR2).

0 = When SMR5 is a logic 1, selects PTGB as the input to the SINC Programmable Counter when SMR5 is a logic 1. The PTGB input may be divided by 3 (see SFR2).

Bit 2-5: **Not Used.**

Bit 6: **PTGA Interrupt Enable.** Control bit.

1 = Enables IRQ6 to be asserted when the PRM A Interrupt Flag (PAM7) is set to a logic 1. Note that PTGB can also generate IRQ6. If both precision time generator interrupts are enabled, the IRQ6 interrupt service subroutine should examine the interrupt flag bit in both PTGA and PTGB mode registers to determine the IRQ6 source.

0 = Disables IRQ6 assertion due to PAM7.

Bit 7: **PTGA Interrupt Flag.** Status bit, read only.. Reading or writing to address $0037 clears the PTGA Interrupt Flag (PAM7).

1 = PTGA accumulator overflow has occurred.

0 = PTGA accumulator overflow has not occurred.

3.11.3. Precision Time Generator B

PTGB consists of five 8-bit registers and a 17-bit pulse accumulator (Figure 3-16). The three input registers—PTGB Buffer (PBB), PTGB Lower Latch (PBLL) and PTGB Upper Latch (PBUL)—are all 8-bit. There are two output registers: an 8-bit PTGB Lower Residue (PBLR) and a 9-bit PTGB Upper Residue (PBUR). The PTGB Accumulator (PBAC) is 17-bits long. Operation is controlled by the PTGB Mode Register (PBM) located at $000F.

Operation of PTGB is identical to PTGA with the exception of register addresses, and the port B interface line (Figure 3-16).

3.11.4. PTGB Mode Register (PBM)

The PTGB Mode Register (PBM) selects the PTGB timer and port options, and controls and reports the PTGB interrupt (Table 3-16). Bits 0, 1, and 6 are cleared by reset and can be set to a logic 1 or 0 by writing to address $001C. All bits can be read by the CPU. Bit 7 is read only.

Table 3-16. Register Bit Assignments - PTGB - 000Ch

Addr.	Function	Bit 7	6	5	4	3	2	1	0
000C	Precision Time Generator B (PTGB) Mode (PBM)	PTG B Interrupt Flag	PTG B Interrupt Enable	Not Used					PTG B Timer Mode
000D	PTGB Buffer (PBB)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
000E	PTGB Upper Latch (PBUL)	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8
000F	PTGB Upper Latch (PBUL)	data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8

Bit 0: **PTGB Timer Mode Select.** Control bit.

1 = PTGB operates as a timer.

0 = PTGB operates as a precision time generator.

Bit 1-5: **Not Used.**

Bit 6: **PTGB Interrupt Enable.** Control bit.

1 = Enables IRQ6 to be asserted when the PRM B Interrupt Flag (PBM7) is set to a logic 1. Note that PTGA can also generate IRQ6. If both precision time generator interrupts are enabled, the IRQ6 interrupt service subroutine should examine the interrupt flag bit in both PTGA and PTGB mode registers to determine the IRQ6 source.

0 = Disables IRQ6 assertion due to PBM7.

Bit 7: **PTGB Interrupt Flag.** Status bit, read only. Reading or writing to address $001F clears the PTGB Interrupt Flag (PBM7).

1 = PTGB accumulator overflow has occurred.

0 = PTGB accumulator overflow has not occurred.

3.11.5. Example Rates

Some examples of standard rates are shown in Table 3-17.

Table 3-17. PTGA or PTGB Generated Standard Data Rates

Rate	Ø2 = 6 MHz		Ø2 =8 MHz		Ø2 = 10 MHz	
	Latch (Hex)	Actual	Latch (Hex)	Actual	Latch (Hex)	Actual
576 KHz	3127	576004.03 Hz	24DD	575988.77 Hz		Hz
500 KHz	2AAB	500015.26 Hz	2000	500000.00 Hz		Hz
250 KHz	1555	249984.74 Hz	1000	250000.00 Hz		Hz
230.4 KHz	13A9	230392.46 Hz	0EBF	230407.71 Hz		Hz
200 KHz	1111	199996.95 Hz	0CCD	200012.21 Hz		Hz
100 KHz	0889	100021.36 Hz	0666	99975.59 Hz		Hz

3.12. USART

The MCU provides a full-duplex serial universal synchronous/asynchronous receiver/transmitter (USART) interface with programmable operating modes and data rates. Serial-to-parallel conversion is performed on data characters received from an external device and parallel-to-serial conversion is performed on data characters received from the MCU internal data bus. A block diagram of the USART is shown in Figure 3-17.

3.12.1. General Operation

Internal timing for both asynchronous and synchronous operation can be referenced to either Timer B or the Precision Timing Generators under software control. Synchronous transmit data (TXD) timing can also be derived externally by an external transmit clock (TXCLK) input on PA3 or an external transmit reference clock (TXREF) input on PA4. Synchronous received data (RXD) timing can be generated from an external receive clock (RXCLK) input on PA7. Note that the direction registers for PA1 through PA7 (PAD1 - PAD7) must be set correctly for the mode selected. Table 3-18 shows how standard data rates can be generated internally using either Timer B or the Precision Timing Generators A and B.

The serial interface registers are located at addresses $0034-$003F (Table 3-19). The CPU may read or write any of the serial interface registers with the exception of Serial Out Divider Latch (SODL) and Serial In Divider Latch (SIDL) which are write only. The Serial Status Register is read-only. Reading and/or writing to some of the registers also causes clearing of interrupt bits or data downloading actions.

3.12.2. Internal Timing

Since internal timing for TXD and RXD is similar, the following discussion covers only TXD timing. It differs only when precision timing generators (PTGs) are selected. TXD uses PTGB and RXD uses PTGA.

Case 1 - Asynchronous Timer B. Whenever short stop bits are selected (SFR0 = 1 or SFR1 = 1), or 5-bit operation is selected (SLCR0 = 0 and SLCR1 = 0), the user must program the SOUT (RXD) Divider Latch (SODL) to $0F. This restriction does not apply to the SIN (TXD) Divider Latch (SIDL).

$$UIB = øB/[(n + 1)(l + 1)]$$

where UIB is the TXD bit rate, øB is the internal clock rate of Timer B, n is the decimal value loaded into the Timer B Latch, and l is the decimal value loaded into the SIDL latch. If ø2 = 6 MHz and the DIV BY 32 option for Timer B is not selected, øB = 6 MHz. If n = 155 ($009B) and l = 15 ($0F), then

$$UIB = 6\ MHz/[(155 + 1)(15 + 1)] = 2403.85\ Hz.$$

Case 2 - Synchronous Timer B. An additional DIV BY 2 is required to generate the 50% duty cycle TXCLK (PA3) required for synchronous operation. Using the same values as in Case 1,

$$UIB = øB/[2(n + 1)(l + 1)] = 6\ MHz/[2\ (155 + 1)(15 + 1)] = 1201.92\ Hz$$

Case 3 - Asynchronous PTGB. Assume that PTGB is generating a rate of 230.4 KHz and that the ø2 internal timing is 6 MHz. From Table 3-18, the PTGB latch is loaded with $13A9. Also, from Table 3-18, PTGB = 230.39246 KHz. If l = 15 ($0F), then:

$$UIB = PTGB/\{[3(l + 1)]\} = 230392.46/[3\ (15 + 1)] = 4799.84\ Hz$$

This assumes that the USART DIV BY 2 mask option (Figure 3-17) is selected. If short stop bits or 5-bit operation is selected, the user must program the SOUT Divider Latch (SODL) to decimal 15 ($0F).

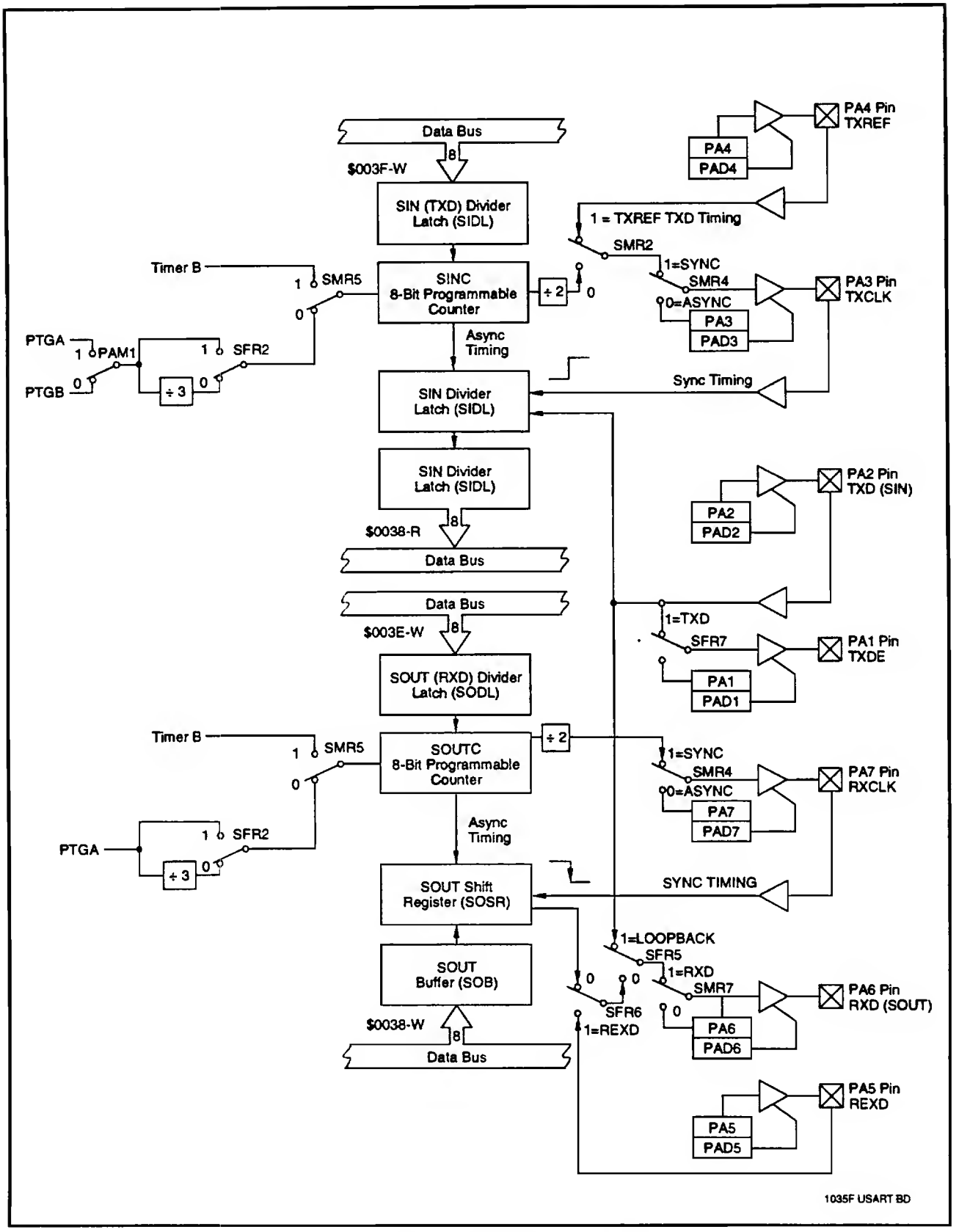


Figure 3-17. USART Block Diagram

Table 3-18. USART Generated Standard Data Rates

Standard Data Rate	Asynchronous, Ø2 = 6 MHz (SIDL = SODL = $06)				Synchronous, Ø2 = 6 MHz			
	Timer B		PTGA and PTGB		TIMB Latch = 009B, Rate = 38461.54 Hz PTG Rate = 230.4 KHz, PTG Latches = $13A9			
	TIMB Latch	Bit Rate	PTG Latch	Bit Rate	SIDL, SODL	Bit Rate	SIDL, SODL	Bit Rate
50 Hz	$1D4B	50.00	$0034	49.59	–	–	–	–
75 Hz	$1387	75.00	$004F	75.34	$FF	75.12	–	–
110 Hz	$0D50	110.00	$0073	109.76	–	–	–	–
150 Hz	$09C3	150.00	$009D	149.73	$7F	150.24	$FF	150.00
300 Hz	$04E1	300.00	$013B	300.41	$3F	300.48	$7F	299.99
600 Hz	$0270	600.00	$0275	599.86	$1F	600.96	$3F	599.98
1200 Hz	$0137	1201.92	$04EA	1199.72	$0F	1201.92	$1F	1199.96
2400 Hz	$009B	2403.85	$09D5	2400.40	$07	2403.85	$0F	2399.92
4800 Hz	$004D	4807.69	$13A9	4799.84	$03	4807.69	$07	4799.84
9600 Hz	$0026	9615.38	$2752	9599.69	$01	9615.38	$03	9599.69
14400 Hz	$0019	14423.08	$3AFB	14399.53	–	–	–	–
19200 Hz	$0013	18750.00	$4EA5	19200.32	$00	19230.77	$01	19199.37
							$00	38398.74

Case 4 - Synchronous PTGB. An extra DIV BY 2 also occurs in the synchronous mode. Using the same setup as in Case 3 and again assuming that the USART DIV BY 2 mask option (Figure 3-17) is selected.

UIB = PTGB/2[3 (I + 1)]= 230392.46/[6 (15 + 1)]= 2399.92 Hz.

3.12.3. USART Registers and Serial Buffer Registers (SB)

The USART control and status registers are shown in Table 3-19. The serial buffer (SB) registers provide double buffering for serial in and serial out data. Both buffers are located at address $0038. The Serial In Buffer is a read-only register and the Serial Out Buffer is a write-only register. All data is shifted least significant bit first. When odd or even parity is used with 5-, 6- or 7-bit character operation, the unused most significant bits in the Serial Out Buffer must be loaded with zeros. Zeros are automatically inserted in the Serial In Buffer.

Table 3-19. Register Bit Assignments: USART - 0038h - 003Fh

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0038	Serial In Buffer (SIB) / Serial Out Buffer (SOB)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0039	Serial Interrupt Register (SIR)	TXD Status Interrupt Flag	TXCLK (PA3) ↑ Interrupt Flag	RXCLK (PA7) ↓ Interrupt Flag	TXCLK (PA3)↑ Interrupt Enable	RXCLK (PA7) ↓ Interrupt Enable	TXD Status Interrupt Enable	RXD Buf Empty Interrupt Enable (BE)	RXD Buf Full Interrupt Enable (BF)
003A	Serial Mode Register (SMR)	RXD On	TXD On	Timing Select 0 = PRGs 1 = TIMB	Sync Mode	TXD Sync Bit	TXREF Clock Select	CTSP/ RLSDP Sync Enable	Not Used
003B	Serial Line Control Register (SLC)	Parity Stuff Bit	Set Break	Stuff Parity	Even Parity	Enable Parity	Two Stop Bits	Word Length Bit 1 (SL1)	Word Length Bit 0 (SL0)
003C	Serial Status Register (SSR)	TXD Parity Bit	RXD Underrun (UR)	RXD Buffer Empty (BE)	TXD Break Int (BI)	TXD Framing Error (FE)	TXD Parity Error (PE)	TXD Overrun Error (OE)	TXD Buffer Full (BF)
003D	Serial Form Register (SFR)	TXD/ TXDE Echo	REXD/ TXD Echo	TXD/ RXD Echo	TIMA Input TXD Test	TXD/ Edge TXD Test	PTG Divide by 3	7/8 Short Stop Bit	3/4 Short Stop Bit
003E	SOUT (RXD) Divider Latch (SODL)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
003F	SIN (TXD) Divider Latch (SIDL)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0

3.12.4. Serial Mode Register (SMR)

The Serial Mode Register (SMR), located at $003A, controls basic serial mode and timing selection. All bits are cleared to zero by reset and can be set or reset by the CPU. Bits 6 and 7 enable the TXD and the RXD modes, respectively, for both asynchronous and synchronous modes of operation. Bits 2-4 control synchronous functions. Bits 2-4 must be set to zero for asynchronous operation.

Bits 0: Not used.

Bits 1: **CTSP/RLSDP Enable.** Control bit.

1 = When TXD Mode is selected (SMR6 = 1), the USART TXD synchronous mode word counter is initialized upon detection of a negative edge on PA0 (normally connected to CTSP). When RXD Mode is selected (SMR7 = 1), the USART RXD synchronous mode word counter is initialized upon detection of a negative edge on PA4 (normally connected to RLSDP).

0 = When set to a logic 0, PA0 and PA4 are not used for USART support.

Bit 2: **Select TXREF Source.** Control bit.

1 = Selects the TXREF on PA4 clock for TXD timing.

0 = Selects internal TXD timing.

Bit 3: **Enable TXD Sync Detect.** Control bit.

1 = TXD data line is monitored for a high-to-low (1-0) character transition. This transition causes serial in word synchronization following the 0 character and resets bit SMR3.

0 = TXD data line is not monitored for a high-to-low (1-0) character transition.

Bit 4: **Sync Mode Select.** Control bit.

1 = Selects synchronous mode operation; PA7 is assigned to RXCLK and PA3 is assigned to TXCLK.

0 = Selects asynchronous mode operation; PA7 and PA3 are general purpose I/O pins.

Bit 5: **Internal Timing Reference.** Control bit.

1 = Selects Timer B for both TXD (SIN) and RXD (SOUT).

0 = Selects PTGA or PTGB for TXD as selected by the SIN (TXD) PTG Select bit in the PTGA Mode Register (PAM1) and selects PTGA for RXD. This allows PTGA to be selected for both TXD and RXD timing thus freeing PTGB for other use. The input from the PTG can be divided by 3 as selected by the PTG Divide by 3 bit in the Serial Form Register (SFR2).

Bit 6: **TXD Mode Select.** Control bit.

1 = TXD timing and shift register are operational. PA2 is assigned to TXD.

0 = PA2 is not assigned to TXD.

Bit 7: **RXD Mode Select.** Control bit.

1 = RXD timing and shift register are operational. PA6 is assigned to RXD.

0 = PA6 is not assigned to RXD.

3.12.5. Serial Interrupt Register (SIR)

The Serial Interrupt Register (SIR) is located at address $0039. The SIR contains five enable bits (bits 0-4) and three interrupt flag bits (bits 5-7). The CPU can read all bits but can write only to the five interrupt enable bits. All SIR bits are cleared to zero by reset. An interrupt enable bit, when set to a logic 1, permits the corresponding condition to assert the associated IRQ.

Bit 0: **TXD (Serial In) Buffer Full Interrupt Enable.** Control bit.

1 = Enables IRQ4 to be asserted when the Serial In Buffer Full bit (SSR0) is a logic 1.

0 = Disables IRQ4 assertion based on SSR0.

Bit 1: **RXD (Serial Out) Buffer Empty Interrupt Enable.** Control bit.

1 = Enables IRQ6 to be asserted when the Serial Out Buffer Empty bit (SSR5) is a logic 1.

0 = Disables IRQ6 assertion based on SSR5.

Bit 2: **Serial In Status Interrupt Enable.** Control bit.

1 = Enables IRQ5 to be asserted when the Serial In Status bit (SIR7) is a logic 1.

0 = Disables IRQ5 assertion based on SIR7.

Bit 3: **PA7 Interrupt Enable.** Control bit.

1 = Enables IRQ6 to be asserted when the PA7 Interrupt Flag (SIR5) is a logic 1.

0 = IRQ6 will not be asserted based on SIR5.

Bit 4: **PA3 Interrupt Enable.** Control bit.

1 = Enables IRQ4 to be asserted when the PA3 Interrupt Flag (SIR6) is a logic 1.

0 = Disables IRQ4 assertion based on SIR6.

Bit 5: **PA7 Interrupt Flag.** Control bit.

1 = A high-to-low transition has been detected on the PA7. RXCLK is connected to this pin during synchronous serial output operation. Writing a 0 to bit 4 of the Clear Interrupt Register clears this bit to a logic 0.

0 = A high-to-low transition has not been detected on the PA7.

Bit 6: **PA3 Interrupt Flag.** Status bit.

1 = A low-to-high transition has been detected on PA3 (SFR3 = 0) or on PA2 (SFR3 = 1). TXCLK is normally connected to PA3 and TXD is normally connected to PA2 in synchronous serial input operation. Writing a 0 to bit 3 of the Clear Interrupt Register clears this bit to a logic 0.

0 = A low-to-high transition has not been detected on PA3 (SFR3 = 0) or on PA2 (SFR3 = 1).

Bit 7: **Serial In Status Interrupt Flag.** Status bit.

1 = Any of the following three flags have been set to a logic 1 in the Serial Status Register (SSR): Framing Error (SSR3), Parity Error (SSR2), or Overrun Error (SSR1). The Serial In Status Interrupt Flag (SIR7) is also set to a logic one whenever the Break Interrupt (SSR4) changes state, i.e., whenever it changes from a 0 to a 1 or from a 1 to a 0. SIR7 is reset by writing to the Serial Status Register (SSR) at address $003C. Writing to $003C clears SIR7 but does not alter any bits in the Serial Status Register (SSR).

0 = None of the following three flags have been set to a logic 1 in the Serial Status Register (SSR): Framing Error (SSR3), Parity Error (SSR2), or Overrun Error (SSR1).

3.12.6. Serial Line Control Register (SLCR)

The Serial Line Control Register (SLCR) at $003B specifies the word length and parity generation and checks in asynchronous mode. Each bit can be set or reset by the CPU. All SLCR bits are initialized to zero by reset. The SLCR must be set to $03 for 8-bit synchronous operation. The Parity Enable bit (SLCR3) must be a zero in the synchronous mode. Both asynchronous and synchronous modes support 5-, 6-, 7-, and 8-bit word lengths.

Bits 0-1: **Word Length.** These two control bits specify the number of bits in each serial in or serial out character. The encoding of bits 0 and 1 is:

Bit 1	Bit 0	Word Length
0	0	5 Bits
0	1	6 Bits
1	0	7 Bits
1	1	8 Bits

Whenever 5 bits are selected in the asynchronous mode the transmitter will generate 1 and 1/2 stop bits. This requires that the SOUT Divider Latch (SODL) be loaded with a decimal 15 ($0F).

Bit 2: **Number Stop Bits.** Control bit.

1 = One and a half stop bits are generated when a 5-bit word length is selected; two stop bits are generated when a 6-, 7-, or 8-bit word length is selected.

0 = One stop bit is generated regardless of word length.

Bit 3: **Enable Parity.** Control bit.

1 = Enables a parity bit to be inserted in the serial out data stream and to be checked for in the serial in data stream. The parity bit is located between the last data bit and the first stop bit.

0 = Disables parity generation and checking.

Bit 4: **Even Parity Select.** Control bit.

1 = When parity is enabled (SLCR3 = 1) and parity stuff is disabled (SLCR5 = 0), a one or zero is automatically inserted into the serial out parity position such that the total number of ones in the data and parity fields is even.

0 = When parity is enabled (SLCR3 = 1) and parity stuff is disabled (SLCR5 = 0), a one or zero is automatically inserted into the serial out parity position such that the total number of ones in the data and parity fields is odd.

Bit 5: **Enable Parity Stuff.** Control bit.

1 = When parity is enabled (SLCR3 = 1) and parity stuff is enabled (SLCR5 = 1), the parity stuff bit value (SLCR7) is copied into the serial out parity position.

0 = The parity stuff bit value (SLCR7) is never copied into the serial out parity position.

Bit 6: **Set Break.** This bit is the Break Control bit. The Break Control bit acts only on RXD and has no effect on the serial in logic.

1 = The received data output (RXD) is forced to the space (logic 0) state.

0 = The break is disabled, i.e., the received data output (RXD) is normal.

Bit 7: **Parity Stuff Bit.** Control bit.

1 = A logic 1 is copied into the RXD parity bit when both parity is enabled (SLCR3 =1) and parity stuff is enabled (SLCR5 = 1).

0 = A logic 0 is copied into the RXD parity bit when both parity is enabled (SLCR3 =1) and parity stuff is enabled (SLCR5 = 1).

3.12.7. Serial Status Register (SSR)

The Serial Status Register (SSR) at $003C provides serial-in and serial-out status to the CPU. It is a read-only register. All bits are initialized by reset to a logic 0, except Serial Out Buffer Empty (SSR5), which is initialized to a logic 1. Writing to the Serial Status Register will not alter any bits in the register, but it will reset the Serial In Status Interrupt Flag (SIR7).

Bit 0: **Serial In Buffer Full (BF).** Status bit.

1 = A complete incoming character, asynchronous or synchronous, has been received and transferred to the Serial In Buffer. This bit is reset to a logic 0 when the CPU reads the Serial In Buffer.

0 = A complete incoming character, asynchronous or synchronous, has not been received and transferred to the Serial In Buffer.

Bit 1: **Overrun Error (OE).** Status bit.

1 = Data in the Serial In Buffer was not read by the CPU before the next received character was transferred into the Serial In Buffer, thereby destroying the previous character. This bit is reset as soon as a received data character can be safely transferred into an empty Serial In Buffer.

0 = A complete incoming character, asynchronous or synchronous, has not been received and transferred to the Serial In Buffer.

Bit 2: **Parity Error (PE).** Status bit.

1 = The received data character does not have the correct even or odd parity, as selected by the Even Parity Select bit (SLCR4). The PE bit is set to a logic 1 upon detection of a parity error. It is reset to a logic 0 as soon as a received data character is found with a correct parity bit.

0 = The received data character has a correct parity bit.

Bit 3: **Framing Error (FE).** Status bit.

1 = The received character did not have a valid stop bit. The FE is set to a logic 1 whenever the stop bit following the last data bit or parity bit is detected as a zero bit (space level). The FE bit is reset as soon as a received data character is found with a correct first stop bit.

0 = The received data character has a correct first stop bit.

Bit 4: **Break Interrupt (BI).** Status bit.

1 = The received data input (serial in) is held in the space (logic 0) state continuously from the start bit to the first stop bit. The BI bit is cleared when a mark bit (logic 1) is detected on TXD.

0 = A mark bit (logic 1) is detected on TXD.

Bit 5: **Serial Out Buffer Empty (BE).** Status bit.

1 = The Serial Out Buffer is empty and ready to accept a new character for transmission. When this bit is true simultaneously with the Serial Out Interrupt Enable bit (SIR1), IRQ6 is asserted. The BE flag is set to a logic 1 when the contents of the Serial Out Buffer are transferred into the Serial Out Shift Register. BE is reset to a logic 0 when the CPU writes to the Serial Out Buffer.

0 = The Serial Out Buffer is not empty

Bit 6: **Serial Out Underrun (UR).** Status bit.

1 = The Serial Out Shift Register has emptied and the Serial Out Buffer has not been reloaded by the CPU. When the UR bit is a logic 1, the Serial Out Shift Register will output mark bits (logic 1). It is reset to a logic 0 when the CPU writes to the Serial Out Buffer.

0 = Serial out underrun has not occurred.

Bit 7: **Serial In Parity Bit.** This bit copies the received serial in parity bit.

3.12.8. Serial Form Register (SFR)

The Serial Form Register (SFR), located at $003D, controls TXD testing and specifies special routing of serial signals and stop bit length adjustment. Bits 5-7 can support local and remote loopback operation. The CPU can read and write all SFR bits except bit 2.

Bits 0-1: **Short Stop Bits.** When set, these bits shorten the width of the final transmitter stop bit. A normal stop bit width occurs when both bits are set to 0. Stop bit control requires the user to load the Serial Out Divider Latch (SODL) with a decimal 15 ($0F).

Bit 1	Bit 0	Final Stop Bit Width 5-bit Word Length, 2 Stop Bit Mode	Final Stop Bit Width All Other Modes
0	0	1/2 Bit	1.0 Bit
0	1	1/4 Bit	3/4 Bit
1	0	3/8 Bit	7/8 Bit
1	1	1/8 Bit	5/8 Bit

Bit 2: **PTG Divide by 3.** Control bit.

1 = The input from the PTG is not divided by 3 before it is routed to the SINC and SOUTC programmable timers when PTG source is selected by the Internal Timing Reference bit in the Serial Mode Register (SMR5).

0 = The input from the PTG is divided by 3 before it is routed to the SINC and SOUTC programmable timers when PTG source is selected by the Internal Timing Reference bit in the Serial Mode Register (SMR5).

Bit 3: **TXD Test - PA3 Edge.** Control bit.

1 = PA2 (TXD) is routed to the PA3 positive edge detect (see SI3).

0 = PA3 (TXCLK) is routed to the PA3 positive edge detect logic.

Bit 4: **TXD Test - Timer A Input.** Control bit.

1 = PA2 (TXD) input is routed to the Timer A input detect logic. SFR3 and SFR4, when set, can be used to easily measure the TXD start pulse width.

0 = PA0 is routed to the Timer A input detect logic.

Bit 5: **Loopback.** Control bit.

1 = PA2 (TXD) is routed back to RXD instead of the normal serial out data.

0 = PA 6 (RXD) will copy the SOUT shift register when SFR6 = 0 or SOP (PA6) when SFR6 = 1.

Bit 6: **Serial Out Passthrough on PA5.** Control bit

1 = The serial out input on PA5 is routed to RXD output on PA6. Note that serial out mode must be selected (SMR7 = 1) and serial loopback must not be selected (SFR5 = 0).

0 = PA5 is a general purpose I/O line.

Bit 7: **Serial In Passthrough on PA1.** Control bit.

1 = PA2 (TXD) input on is routed to the serial in output on PA1.

0 = PA1 is a general purpose I/O line.

3.12.9. Serial Out (RXD) Divider Latch (SODL)

The Serial Out Divider Latch (SODL) is a CPU write-only register at $003E that controls operation of the RXD timing (Figure 3-17). The SODL must be loaded with the value corresponding to the desired data. Table 3-18 contains standard values for asynchronous and synchronous operation. Note that the SODL must be loaded with $0F for asynchronous operation whenever Short Stop Bit operation is required. Each time that the CPU writes to the SODL, the new latch value is downloaded to the counter.

3.12.10. Serial In (TXD) Divider Latch (SIDL)

The Serial In Divider Latch (SIDL) is also a CPU write-only register at $003F that controls operation of the TXD timing (Figure 3-17). Its operation is similar to that of the SODL, except that it is not restricted to a value of $0F for asynchronous operation.

3.13. DUAL PORT RAM - GP /16550A INTERFACE

The Dual Port RAM interface can be configured to operate in one of three modes:

1. General purpose mode (GP mode) 13-byte Dual Port RAM (DP RAM) with host handshake register, GP FIFO status register, and transmit/receive 16-byte FIFOs.

2. 16450 register compatible mode.

3. 16550A register compatible mode with transmit/receive 16-byte FIFOs.

The MCU/host interface diagram is shown in Figure 3-18. Signal equivalence between the MCU and each mode is shown in Table 3-20.

The selectable host bus interface provides an 6500 or RDP/WTP bus compatible interface between the MCU and a host microprocessor (Figure 3-18). This interface allows the MCU to act like a standard peripheral device connected to the host bus under control of the host processor. Under MCU software control, this interface can be a general purpose user-defined interface or an emulation of the 16450/16550A UART interface.

Built-in hardware registers and control signals allow a 16450/6550 UART compatible interface to be presented to the host bus. More supporting 16450/6550 interface functions are provided in MCU hardware than in earlier models in order to relieve the MCU application firmware from having to provide some time critical and overhead functions when servicing the interface. Supporting MCU application firmware functions are required to fully implement the 16450/16550A interface, however.

3.13.1. Host Bus Interface Signals and Registers

When the host bus interface is selected (HCR2 = 1), the following host bus signals are supported instead of the general purpose I/O lines on ports A (4 lines), B (2 lines), C (8 lines), and D (7 lines):

8-bit bidirectional data lines (HD0-HD7)

4-bit address inputs (HA0-HA3), HA3 remains a GP I/O in the 16450/16550A mode.

1 chip select input (HCSP)

2 bus timing inputs (HWTP and HRDP, or Hø2 and HR/WP)

1 host interrupt output (HINT)

2 data ready outputs (RXRDY and TXRDY) if 16450/16550 DP RAM is selected (HCR1 = 1), FIFO is enabled (FCR0 = 1), and serial mode is not selected (SMR7 = 0 and SMR4 = 0)

2 data acknowledge inputs (RXACKP and TXACKP) if 16450/16550 DP RAM is selected (HCR1 = 1), FIFO is enabled (FCR0 = 1), and serial mode is not selected (SMR7 = 0 and SMR4 = 0)

The host bus waveforms and timing are described in Section 5.

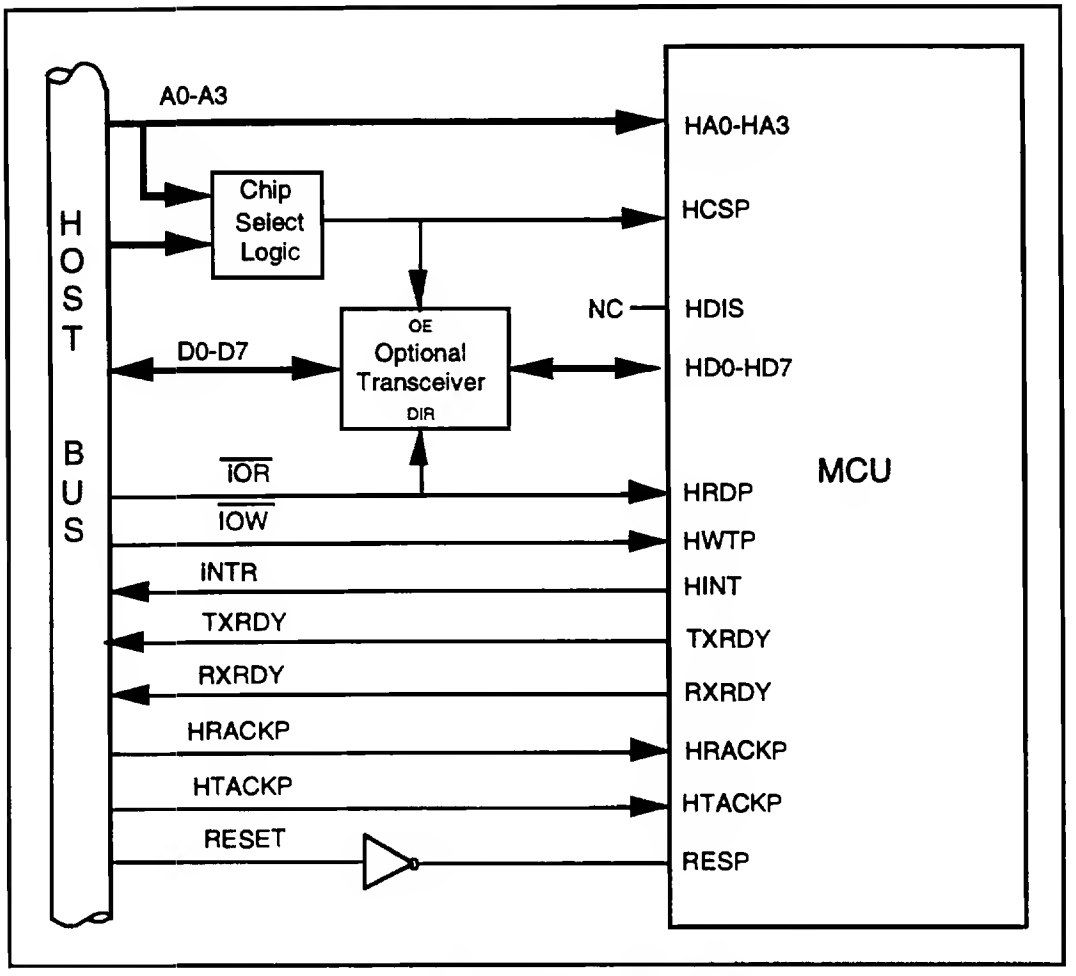


Figure 3-18. MCU/Host Interface

Table 3-20. MCU Dual Port RAM Signal Equivalence

MCU Port Name	MCU Signal Name	GP RAM Mode	16450 Mode	16550A Mode
PB6	HDIS*	HDIS	DDIS	DDIS
PB7	HINT	HINT	INTR	INTR
PC0-PC7	HD0-HD7	HD0-HD7	D0-D7	D0-D7
PD0-PD2	HA0-HA2	HA0-HA2	A0-A2	A0-A2
PD3	HA3	HA3	-	A3
PD4	HCSP	HCSP	CS2P	CS2P
PD5	HWTP	HWTP	WTP	WTP
PD6	HRDP	HRDP	RDP	RDP
PA3	HTACKP	-	HTACKP	HTACKP
PA4	HRACKP	-	HRACKP	HRACKP
PA5	TXRDY	-	TXRDY	TXRDY
PA6	RXRDY	-	RXRDY	RXRDY
RESP	RESP	RESP	NR	MR

* Not recommended for use.

3.13.2. General Purpose Interface

See Table 3-21 for a memory map of the general purpose dual port RAM. TXRDY, RXRDY, HTACKP, and HRACKP are **not** supported in the GP mode. The FIFO Status Register (FSR) and FIFO Interrupt Enable Register (FIER) have the same function as in the 16450/16550A mode. The host cannot access the FIFO Status and FIFO Interrupt Enable registers. In this mode, the Transmitter and Receiver FIFOs (TX FIFO and RX FIFO) are each 16 bytes deep and 8 bits wide. The host can only read data from the RX FIFO and can only write data to the TX FIFO. Conversely, the MCU can only write data to the RX FIFO and can only read data from the TX FIFO.

Both host and MCU can read and write to SP RAM locations $1 through $F. Simultaneous MCU write and host write to these locations are allowed but should be avoided. This can best be accomplished by using the Host Handshake Register (HHR). Tables 3-21 and Table 3-23 provide information on how the HHR can be polled or used to control both host and MCU interrupts. The GP FIFO Status Register (GPFS) provides the host with FIFO control and status information when using the TX FIFO and RX FIFO.

When the GP mode is selected, RUCLK OFF (FIE4) should be enabled. This locks TUPTR to TXPTR and RUPTR to RXPTR. Both timeout counters remain functional.

Table 3-21. MCU Host Bus Interface Memory Map - GP RAM Mode (HCR1 = 0)

MCU Address (Hex)	Function	Host Address (Hex)
0020	RX FIFO Buffer (Host Read/MCU Write)	0
0020	TX FIFO Buffer (Host Write/MCU Read)	0
0021	SP RAM 1	1
0022	SP RAM 2	2
0023	SP RAM 3	3
0024	SP RAM 4	4
0025	SP RAM 5	5
0026	SP RAM 6	6
0027	SP RAM 7	7
0028	SP RAM 8	8
0029	SP RAM 9	9
002A	SP RAM A	A
002B	SP RAM B	B
002C	SP RAM C	C
002D	SP RAM D	D
002E	GP FIFO Status (GPFS)	E
002F	Host Handshake Register (HHR)*	F
0030	FIFO Status Register (FSR)	Not Accessible by Host
0031	FIFO Interrupt Enable (FIER)	Not Accessible by Host
0032	Host Control Register (HCR)	Not Accessible by Host

* Host Handshake Register:
Host Interrupt = (HE1)(HF1) + (HE2)(HF2)
MCU Interrupt = (CE1)(CF1) + (CE2)(CF2)

3.13.3. 16550A Interface Mode and FIFOs

When the FIFO mode is selected (HCR1 = 1, HCR2 = 1, and FCR0 = 1), the MCU provides a 16550A register compatible interface mode. The 16550A interface registers are described in Tables 3-22 and 3-23. The 16550A interface uses the hardware signals shown in Figure 3-18 except the A3 signal is not used. The operation of the FIFOs is illustrated in Figures 3-19 to 3-21.

FIFO UART Timing Simulator. The FIFO UART Timing Simulator (see Figure 3-20) provides both an RUCLK and an TUCLK to simulate the 16550 UART word rate. These rates are driven by either TIMB or PTGB through divide by 16 counters.

RUCLK is automatically re-synchronized to one sixteenth of the word rate by either the termination of Freeze (FSR5) or by the first MCU RX FIFO write to occur when both RUPTR and RXPTR are empty. TUCLK will be re-synchronized by the first host write to the TX FIFO to occur when both TUPTR and TXPTR are empty. when the host writes to an empty FIFO the first TUCLK does not occur until a UART delay has occurred. Note that host can read data from the RX FIFO and write data to the TX FIFO faster than the baud rate by monitoring the 16550 interface interrupt flags rather than waiting for the baud clock to elapse. This timer "override" feature allows much faster data throughput rates than normal baud timing. Note also that when using this technique, the buffers must be completely emptied for the MCU to properly re-synchronize upon receipt of the next byte at the word rate.

TX FIFO. In the 16550A mode, the host can burst data into the TX FIFO Buffer until the TX FIFO is full. This can be accomplished under host control or by a supported DMA mode using TXRDY and TXACKP lines.

Both TXPTR and TUPTR advance together as the host fills the TX FIFO. The CPU unloads the TX FIFO using TXPTR. TUPTR simulates the UART by decrementing the TUCLK rate until TX FIFO is emptied. This causes THRE to be asserted which signals the host that the TX FIFO is empty. In the event the host does not write to the TX FIFO before the next TUCLK, TEMT is asserted. Two interrupt request conditions with enables are provided to the CPU: TCDA (TX FIFO data available) and TCHF (TX FIFO half full). If the TX FIFO contains between one and seven bytes of data and TCHF is enabled and neither the host or CPU have accessed the TX FIFO for three or four TUCLK intervals, a TCTO character time-out interrupt is provided to the CPU. Both TXPTR and TUPTR are reset whenever the TX FIFO bit (FCR2) is set to a 1 by the host, or whenever FIER3 is set to a 1 by the CPU, or, for the C40 and L39 only, whenever the host changes the state of FCR0. These reset controls are self clearing.

RX FIFO. In the 16550A mode, the host can burst read data from the RX FIFO Buffer until the RX FIFO is empty. This can be accomplished under host control or by a supported DMA mode using RXRDY and RXACKP lines.

The RX FIFO contains 16 eleven-bit words. Each eleven bits contain 8 bits of data and a 3-bit error field. The error field is established by the CPU loading PE, FE and BI data into the RX FIFO Status Register (FSR) bits 2 - 4. The next IWRX clock causes the error data stored in the FSR to be inserted together with CPU data into the RX FIFO. The error bits PE, FE, and BI in the FSR can be cleared only by the CPU writing 0s to these bit positions. IWRX also processes or shifts older information contained in the RX FIFO. RXPTR always points to the oldest unread data and error bits contained in the RX FIFO. The RX FIFO error indication (LSR7) is asserted whenever there is an unread error in the RX FIFO error field. Both RXPTR and RUPTR are reset whenever the RX FIFO bit (FCR1) is set to a 1 by the host, or FIER2 is set to a 1 by the MCU, or whenever the host changes the state of FCR0. These reset control bits are self clearing.

RXPTR permits the MCU to burst-fill the error and data fields of the RX FIFO. The error field can be ignored by leaving BI, FE and PE in the FIFO Status Register in their zero state. Two CPU interrupt request conditions with separate enables are provided: RCEMT (RX FIFO empty) and RCHE (RX FIFO half empty) (see Figure 3-21). Bit 0 of the GP status register provides an RX FIFO full indication.

The RXPTR pointer goes from empty to not empty when the CPU writes the first data into an empty RX FIFO. The RX FIFO is considered empty when both RXPTR and RUPTR are at zero. When the CPU writes to an empty RX FIFO, the RUCLK timer is resynchronized. The RUPTR transition from empty to not empty is delayed until the first RUCLK interval following the resynchronization. At this point the RUPTR pointer increments at the RUCLK rate until it reaches the value in RXPTR. If the trigger level has not been reached and RUPTR reaches the level in RXPTR, a receiver time-out event is started. Once started, if neither an IWRX or ERRX occur for three or four RUCLK intervals, an RXTO time out event is used to generate a host character time-out interrupt (IIR3). The host can set the RX FIFO trigger level at 1, 4, 8 or 14 using the FIFO Control Register.

3.13.4. 16450 Interface Mode and FIFOs

When the 16450 mode is selected (HCR1 = 1, HCR2 = 1, and FCR0 = 0), the MCU provides a 16450 register compatible interface mode. The 16450 interface uses the hardware signals shown in Figure 3-18 except the A3, TXRDY, RXRDY, HTACKP, or HRACKP signals are not used. The 16450 interface operation is a subset of the 16550A interface operation.

The 16450 interface registers are described in Tables 3-22 and 3-23. All registers are initialized as shown in Table 3-24 by MCU reset. The CPU has access to both FIFOs while in the 16450 mode.

FIFO UART Timing Simulator. The FIFO UART Timing Simulator (see Figure 3-20) provides both an RUCLK and an TUCLK to simulate the 16450 UART word rate. These rates are driven by either TIMB or PTGB through divide by 16 counters.

RUCLK is automatically re-synchronized to one sixteenth of the word rate by either the termination of Freeze (FSR5) or by the first MCU RX FIFO write to occur when both RUPTR and RXPTR are empty. TUCLK will be re-synchronized by the first host write to the TX FIFO to occur when both TUPTR and TXPTR are empty. Note that host can read data from the RX FIFO and write data to the TX FIFO faster than the baud rate by monitoring the 16550 interface interrupt flags rather than waiting for the baud clock to elapse. This timer "override" feature allows much faster data throughput rates than normal baud timing. Note also that when using this technique, the buffers must be completely emptied for the MCU to properly re-synchronize upon receipt of the next byte at the word rate.

TX FIFO. In the 16450 mode, the host is limited to writing a single byte of data with each THRE interrupt. When the host writes to an empty TX FIFO the THRE bit is cleared and a TUCLK is generated; after a short delay THRE is set true. The host responds by sending new data to the TX FIFO Buffer which then clears THRE. TUCLK will cause THRE to be asserted one word time after the first host write to the empty TX FIFO Buffer. In the event that THRE has not been reset prior to the next TUCLK, TEMT is asserted. TUPTR simulates the responses of a 16450 UART while TXPTR provides normal TX FIFO support for the MCU.

RX FIFO. In the 16450 mode, the CPU has full use of the RX FIFO. The RX FIFO trigger level is set at one. Following a CPU write to an empty RX FIFO Buffer, the DR flag is not set until after an RUCLK delay. A host RX FIFO read, ERRX, decrements RUPTR and resets DR. As long as the RX FIFO contains data, the DR flag will continue to be set following the next RUCLK delay.

Bit 0 of the GP status register provides an RX FIFO full indication.

The RXRDY and RXTO (time-out) signals are non-operational in the 16450 mode.

Table 3-22. MCU Host Bus Interface Memory Map - 16450/16550 Mode

MCU (Internal) Access				Host (External) Access			
Addr.	HCR1	Read	Write	DLAB	Addr.	Read	Write
0020	X	TX FIFO Buffer	RX FIFO Buffer	0	0	RX FIFO Buffer	TX FIFO Buffer
0021	1	Line Status Register (LSR)	LSR1 only	X	5	Line Status Register	Line Status Register
0022	1	Modem Status Register (MSR)	Modem Status Register	X	6	Modem Status Register	Modem Status Register
0023	X	Line Control Register (LCR)	Line Control Register	X	3	Line Control Register	Line Control Register
0024	1	Modem Control Register (MCR)	—	X	4	Modem Control Register	Modem Control Register
—	—	—	—	0	1	Interrupt Enable Register (IER)	Interrupt Enable Register
—	—	—	—	X	2	Interrupt Identifier Register (IIR)	—
0025	1	FIFO Control Register (FCR)	—	X	2	—	FIFO Control Register
0026	X	SP RAM 6	SP RAM 6	—	—	—	—
0027	X	SP RAM 7	SP RAM 7	X	7	Scratch Register	Scratch Register
0028	X	Divisor Latch LSB (DLL)	Divisor Latch LSB	1	0	Divisor Latch LSB	Divisor Latch LSB
0029	X	Divisor Latch MSB (DLM)	Divisor Latch MSB	1	1	Divisor Latch MSB	Divisor Latch MSB
002A	X	SP RAM A	SP RAM A	—	—	—	—
002B	X	SP RAM B	SP RAM B	—	—	—	—
002C	X	SP RAM C	SP RAM C	—	—	—	—
002D	X	SP RAM D	SP RAM D	—	—	—	—
002E		GP FIFO Status	GPFS3 Only	—	—	—	—
002F	X	Host Handshake Register (HHR)	Host Handshake Register	—	—	—	—
0030	X	FIFO Status Register (FSR)	FIFO Status Register	—	—	—	—
0031	X	FIFO Interrupt Enable (FIER)	FIFO Interrupt Enable	—	—	—	—
0032	X	Host Control Register (HCR)	Host Control Register	—	—	—	—

Table 3-23. Register Bit Assignments: 0020h - 0032h

Addr.	Function	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0020	RX FIFO Buffer (Host Read/MCU Write)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0020	TX FIFO Buffer (Host Write/MCU Read)	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0
0021	Line Status Register (LSR)	RX FIFO Error	TX FIFO Empty (TEMT)	TX FIFO Empty (THRE)	Break Interrupt (BI)	Framing Error (FE)	Parity Error (PE)	Overrun Error (OE)	RX Data Ready (DR)
0022	Modem Status Register (MSR)	Data Carrier Detect (DCD)	Ring Indicator (RI)	Data Set Ready (DSR)	Clear to Send (CTS)	Delta Data Carrier Detect (DDCD)	Trailing Edge of Ring Indicator (TERI)	Delta Data Set Ready (DDSR)	Delta Clear to Send (DCTS)
0023	Line Control Register (LCR)	DLAB	Set Break	Stick Parity	Even Parity	Parity Enable	Number Stop Bits	Word Length (WLS1)	Word Length (WLS0)
0024	Modem Control Register (MCR)	0	0	0	Loop	Out 2	Out 1	Request to Send (RTS)	Data Terminal Ready (DTR)
0025	FIFO Control Register (FCR)	RCVR Trigger MSB	RCVR Trigger LSB	Reserved	Reserved	DMA Mode Select	TX FIFO Reset	RX FIFO Reset	FIFO Enable
Host Rd/Wt @ x1	Interrupt Enable Register (IER)	0	0	0	0	Modem Status Interrupt Enable (EDSSI)	RX Line Status Interrupt Enable (ELSI)	TX Hold Empty Interrupt Enable (ETBEI)	RX Data Avail Interrupt Enable (ERBFI)
Host Read @ x2	Interrupt Identifier Register (IIR)	FIFO Enable (FCR0)	FIFO Enable (FCR0)	0	0	Interrupt ID Bit 2	Interrupt ID Bit 1	Interrupt ID Bit 0	0 if Interrupt Pending
0027	Scratch Register (SCR)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0028	Divisor Latch LSB	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0029	Divisor Latch MSB	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
002E	GP FIFO Status (GPFS)	Tx Empty Interrupt Enable	Tx Empty Interrupt Flag	Tx FIFO Half Empty	Rx Trig Level Sel (RTL1)	Rx Trig Level Sel (RTL0) [GP Mde] TuClk Off [16550]	Rx Trig Level Int Enable	Rx Trig Level Int Flag	RCNE [GP Mde] Rx FIFO Data Avail [16550]
002F	Host Handshake Register (HHR)	Controller Intrpt 1 Flag (CF1)	Controller Intrpt 2 Flag (CF2)	Host Intrpt 1 Flag (HF1)	Host Intrpt 2 Flag (HF2)	Controller Intrpt 1 Enable (CE1)	Controller Intrpt 2 Enable (CE2)	Host Intrpt 1 Enable (HE1)	Host Intrpt 2 Enable (HE2)
0030	FIFO Status Register (FSR)	TX Half Full Flag (TCHF)	Data Avail (TCDA)	RX FIFO Freeze	Receiver Break Interrupt (BI)	Receiver Framing Error (FE)	Receiver Parity Error (PE)	RX FIFO Empty Flag (RCEMT)	RX FIFO Half Emty Flag (RCHE)
0031	FIFO Interrupt Enable Register (FIER)	TX FIFO Half Full Interrupt Enable (TCHFE)	Data Avail Interrupt Enable (TCDAE)	UART Timing Select	RUCLK Off	TX FIFO Reset	RX FIFO Reset	RX FIFO Empty Interrupt Enable (RCEMTE)	RX FIFO Half Emty Interrupt Enable (RCHEE)
0032	Host Control Register (HCR)	TX FIFO Interrupt	Mode Control Write Flag	Line Control Write Flag	Divisor Latch Write Flag	RX FIFO Interrupt	Host Mode Select	16450/ 16550 Mode	16450/ 16550 Interrupt Enable

Table 3-24. 16550 Register Initialization

Register	Bit							
	7	6	5	4	3	2	1	0
Line Status Register (LSR)	0	1	1	0	0	0	0	0
Modem Status Register (MSR)	X	X	X	X	0	0	0	0
Line Control Register (LCR)	X	X	X	X	X	X	X	X
Modem Control Register (MCR)	0	0	0	0	0	0	0	0
Interrupt Enable Register (IER)	0	0	0	0	0	0	0	0
Interrupt Identifier Register (IIR)	0	0	0	0	0	0	0	1
FIFO Control Register (FCR)	0	0	X	X	0	0	0	0
GP FIFO Status (GPFS)	0	1	1	0	0	0	0	0
Host Handshake Register (HHR)	0	0	0	0	0	0	0	0
FIFO Status Register (FSR)	0	0	0	0	0	0	0	0
FIFO Interrupt Enable (FIER)	0	0	0	0	0	0	0	0
Host Control Register (HCR)	0	0	0	0	0	0	0	0

TX FIFO Block Diagram

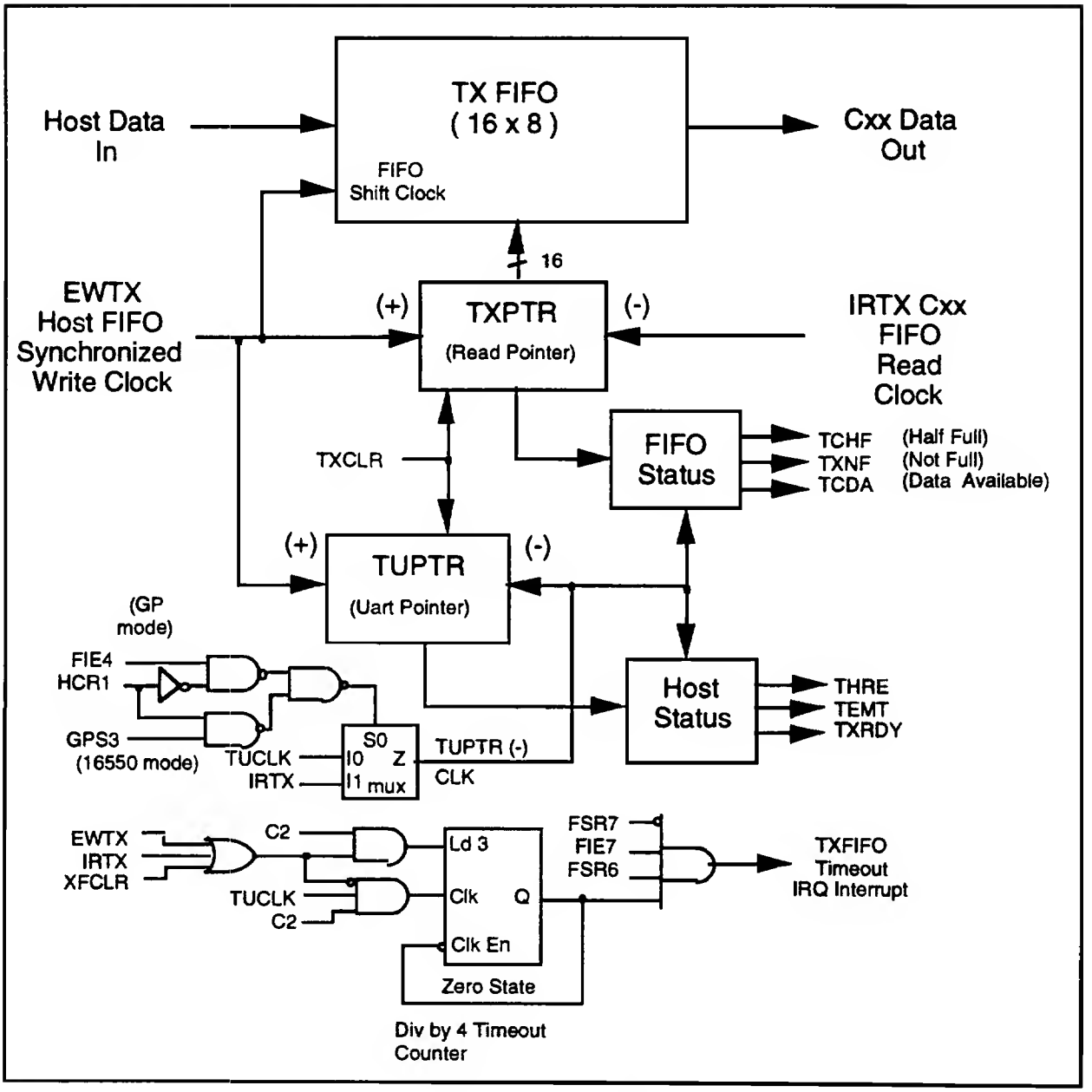


Figure 3-19. TX FIFO Block Diagram

FIFO UART Timing Simulator

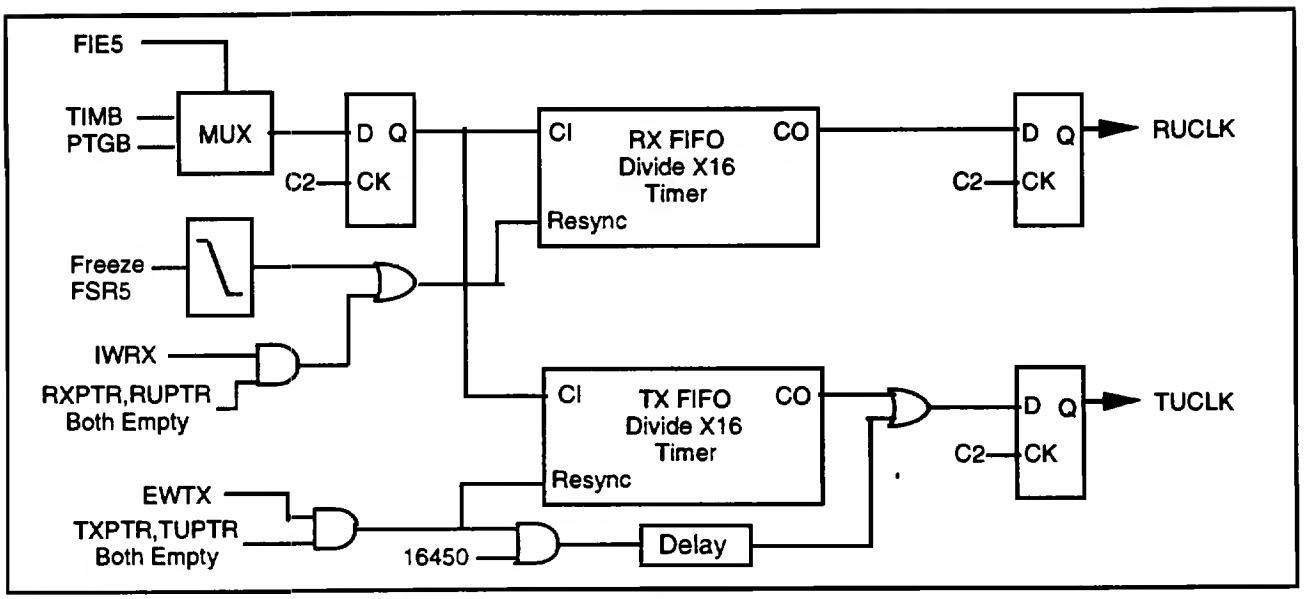


Figure 3-20. FIFO UART Timing Simulator

RX FIFO Block Diagram

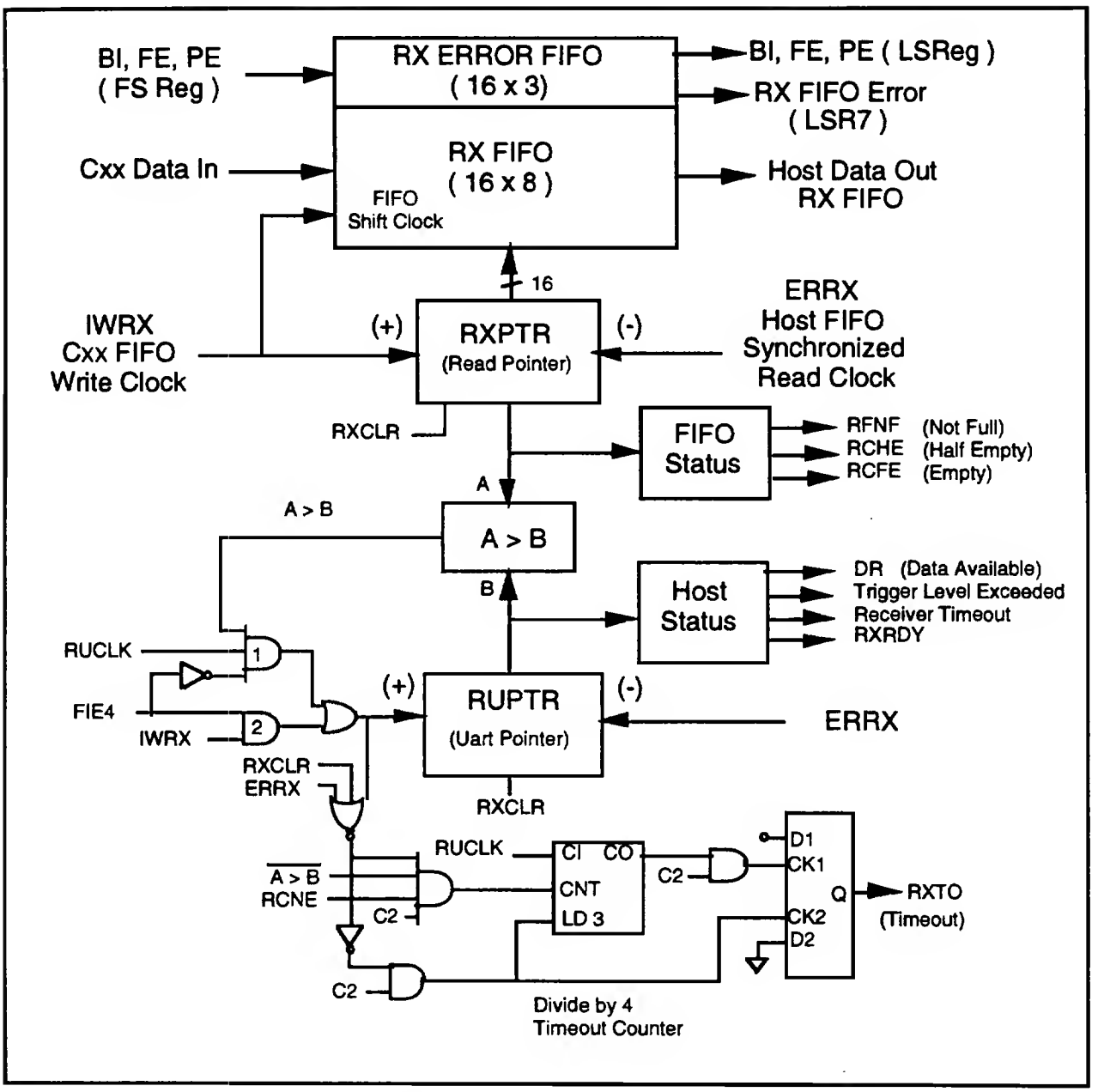


Figure 3-21. RX FIFO Block Diagram

3.13.5. Host Control Register (HCR)

The Host Control Register (HCR) controls and monitors the operation of the 16450/16550 interface. This register cannot be accessed by the host. All bits can be read by the CPU Bits 0-2 can be written to either state by the CPU. Bits 4-6 can be cleared by the CPU (by writing 0s to the corresponding bit locations) but cannot be set by the CPU. All HCR bits are cleared by MCU reset.

Bit 0: **16450/16550 Interrupt Enable**. Control bit.

1 = Enables IRQ3 to be asserted if any interrupt flag in HCR4 through HCR6 is set.

0 = Disables IRQ3 assertion due to any interrupt flag in HCR4 through HCR6.

Bit 1: **16450/16550 DP RAM Mode.** Control bit.

1 = Selects 16450/16550 dual port (DP) RAM mode if the host mode is enabled (HCR2 = 1).

0 = Selects General Purpose DP RAM mode if the host mode is enabled (HCR2 = 1).

Bit 2: **Host Mode Enable**. Control bit.

1 = Enables I/O port host mode (PA3-PA6, PB6-PB7, PC0-PC7, and PD0-PD6 operate as dedicated host I/O pins). This bit must be set for DP RAM to operate in GP DP RAM or 16450/16550 DP RAM mode (see HCR1).

0 = Disables I/O port host mode (PA3-PA6, PB6-PB7, PC0-PC7, and PD0-PD6 operate as general purpose I/O lines).

Bit 3: **RX FIFO Interrupt**. Status bit.

1 = IRQ3 has been asserted due the RX FIFO Half Empty Flag (FSR0) being set or the RX FIFO Empty Flag (FSR1) being set. This bit is set whenever the following logic term is true:

(FSR0)(FIER0) + (FSR1P)(FIER1)

0 = IRQ3 has not been asserted due FSR0 or FSR1 being set.

Bit 4: **Divisor Latch Write Flag**. This bit is set when the host writes to either the Divisor Latch LSB or Divisor Latch MSB register. IRQ3 is asserted if HCR0 is a 1 and this bit is a 1. This bit is cleared by the CPU writing a 0 to this bit position; writing a 1 has no effect.

Bit 5: **Line Control Register Write Flag**. This bit is set when the host writes to the Line Control Register. IRQ3 is asserted if HCR0 is a 1 and this bit is a 1. This bit is cleared by the CPU writing a 0 to this bit position; writing a 1 has no effect.

Bit 6: **Mode Control Register Write Flag**. This bit is set when the host writes to the Mode Control Register. IRQ3 is asserted if HCR0 is a 1 and this bit is a 1. This bit is cleared by the CPU writing a 0 to this bit position; writing a 1 has no effect.

Bit 7: **TX FIFO Interrupt**. Status bit.

1 = IRQ3 has been asserted due the TX FIFO Data Available Flag (FSR6) being set, TX FIFO Half Full Flag (FSR7) being set, or TX FIFO Time-out (FSR7)(FSR6)(TX FIFO Time-out) occurring. A TX time out occurs when neither a EWTX, IRTX or a TXCLR occur for three or four TUCLK intervals. The TX time out is cleared by EWTX, IRTX or a TXCLR. This bit is set whenever the following logic term is true:

(FSR6)(FIER6) + (FSR7)(FIER7) + (FSR7P)(FSR6)(FIER7)(TX Time out).

0 = IRQ3 has not been asserted due FSR6, FSR7, or TX Time out.

3.13.6. Interrupt Enable Register (IER)

The host can read or write bits 0 through 3. The CPU can neither read nor write these bits. These bits are cleared by MCU reset.

The IER enables five types of interrupts that can separately assert the HINT output signal. A selected interrupt can be enabled by setting the corresponding enable bit to a logic 1, or disabled by setting the corresponding enable bit to a logic 0. Disabling all interrupts inhibits the Interrupt Identifier Register (IIR) and inhibits assertion of the HINT output.

The typical use of these bits in a 16450/16550A interface application is:

Bit 0: **Enable Receiver Buffer Full Interrupt (ERBFI) and Character Time-out in FIFO Mode.** Control bit.

- 1 = Enables assertion of the HINT output when the Data Ready bit in the Line Status Register (LSR0) is set to a logic 1 or character time-out occurs in FIFO mode.
- 0 = Disables assertion of HINT due to LSR0 or character time-out.

Bit 1: **Enable Transmitter Buffer Empty Interrupt (ETBEI).** Control bit.

- 1 = Enables assertion of the HINT output when the Transmitter Empty bit in the Line Status Register (LSR5) is set to a logic 1.
- 0 = Disables assertion of HINT due to LSR5.

Bit 2: **Enable Receiver Line Status Interrupt (ELSI).** Control bit.

- 1 = Enables assertion of the HINT output when bit 1, 2, 3, or 4 in the Line Status Register (LSR) is a logic 1.
- 0 = Disables assertion of HINT due to setting of any of LSR1-LSR4 bits.

Bit 3: **Enable Modem Status Interrupt (EDSSI).** Control bit.

- 1 = Enables assertion of the HINT output when bit 0, 1, 2, or 3 in the Modem Status Register (MSR) is a logic 1.
- 0 = Disables assertion of HINT due to setting of any of the MSR0-MSR3 bits.

Bits 4-7: **Not used.** Always 0.

3.13.7. Interrupt Identifier Register (IIR)

The host can read all bits but cannot write to the IIR. All bits are controlled by MCU hardware. The IIR is not accessible to the CPU.

The IIR identifies the existence and type of five prioritized pending interrupts. Four priority levels are set to assist interrupt processing in the host. When accessed, the IIR freezes the highest priority Interrupt pending and acknowledges no other interrupts until the particular interrupt is serviced by the host.

Bits 4 and 5 are not used and always read low. Bits 6 and 7 copy the state of FCR0, the host-controlled FIFO Enable bit. The IIR is initialized to 001b by MCU reset.

Bit 0: **Interrupt Pending.** Status bit. This bit can be used in a hardwired prioritized or polled environment to indicate whether an interrupt is pending. If an interrupt is pending. the IIR contents can be used as a pointer to the appropriate interrupt service routine in the host.

1 = An interrupt is not pending.

0 = An interrupt is pending.

Bits 1-3: **Highest Priority Pending Interrupt.** These three bits identify the highest priority pending interrupt when bit 0 is a logic 0:

Bit 3	Bit 2	Bit 1	Priority Level	Pending Interrupt
0	1	1	1 (highest)	Receiver Line Status
0	1	0	2	Receiver Buffer Full
1	1	0	2	Character Timeout
0	0	1	3	Transmitter Buffer Empty
0	0	0	4	Modem Status

Bits 4-5: **Not used.** Always 0.

Bits 6-7: **FIFO Mode.** These two bits copy FCR0.

3.13.8. Line Control Register (LCR)

All Line Control Register (LCR) bits can be read or written to either by the host or the CPU. The LCR is not initialized by MCU reset.

HCR bit 5 is automatically set to a logic 1 whenever the host writes to the LCR, providing a flag to the CPU that the LCR has been updated. User-provided software must read and interpret the LCR contents.

Bit 7 is duplicated as a separate hardware latch and provides the DLAB address extension bit used for register access. This bit cannot be written to by the CPU.

The typical use of these bits in a 16450/16550 interface application is:

Bits 0-1: **Word Length Select (WLS0 and WLS1).** These two bits specify the number of bits in each transmitted or received serial character (word):

Bit 1	Bit 0	Word Length	No. of Stop Bits (LCR2 = 1)
0	0	5 Bits	1 1/2
0	1	6 Bits	2
1	0	7 Bits	2
1	1	8 Bits	2

Bit 2: **Number Stop Bits (STB).** This control bit specifies the number of stop bits in each transmitted and received in each serial character. The receiver logic checks the first stop bit only regardless of the number of stop bits selected.

1 = The number of stop bits generated depends on the word length (see bits 0 and 1).

0 = One stop bit is generated.

Bit 3: **Parity Enable (PEN).** Control bit.

1 = Enables parity generation and checking. An even or odd Parity bit is generated in the transmitted data or checked in the received data in accordance with Even Parity Select (LCR4) and Stick Parity (LCR5) bits. The parity bit is located between the last data bit and the first Stop bit.

0 = Disables parity generation and checking.

Bit 4: **Even Parity Select (EPS).** Control bit.

1 = Selects even parity. When parity is enabled (LCR3 = 1) and stick parity is disabled (LCR5 = 0),an even number of total number of ones in the data and parity fields is generated or checked.

0 = Selects odd parity. When parity is enabled (LCR3 = 1) and stick parity is disabled (LCR5 = 0), an odd number of total number of ones in the data and parity fields is generated or checked.

Bit 5: **Stick Parity.** Control bit.

1 = Enables stick parity. When parity is enabled (LCR3 = 1), the parity bit is generated and check in accordance with the inverted state of LCR4 (parity bit = 0 when LCR4 = 1; parity bit = 1 when LCR4 = 0).

0 = Disables stick parity.

Bit 6: **Set Break.** Break Control bit. The Break Control bit acts only on RXD and has no effect on the transmitted data.

1 = Sets break. The received data output is forced to the space (logic 0) state.

0 = Disables break.

Bit 7: **Divisor Latch Access Bit (DLAB).** Control bit.

1 = Enables access to the divisor latches of the baud generator during a read or write operation.

0 = Enables access to the RX FIFO Buffer, TX FIFO Buffer, or Interrupt Enable Register during a read or write operation.

3.13.9. Modem Control Register (MCR)

Bits 0-4 of the Modem Control Register (MCR) can be read or written to by the host. The CPU can only read the MCR. When the host mode is selected, the HINT output driver is placed in a three-state mode when either Out2 is false (MCR3 = 0) or Loop is selected (MCR4 = 1). In addition, when the Loop is selected, the operation of the MSR is modified as described under Modem Status Register. The MCR is cleared by MCU reset.

The typical use of these bits in a 16450/16550A interface application is:

Bit 0: **Data Terminal Ready (DTR).** This bit controls the Data Terminal Ready (DTR) function.

1 = DTR is on.

0 = DTR is off.

Bit 1: **Request to Send (RTS).** This bit controls the Request to Send (RTS) function.

1 = RTS is on.

0 = RTS is off.

Bit 2: **Output 1.** Control bit. This bit is used in local loopback (see MCR4).

Bit 3: **Output 2.** Control bit.

1 = HINT is enabled.

0 = HINT is in the high impedance state.

Bit 4: **Loop.** Control bit.

1 = Enables loop. The diagnostic mode is selected and the following occurs:

1. Data written to the TX FIFO Buffer is looped back to the RX FIFO Buffer.

2. The contents of MCR bits 0-3 are internally connected to the four MSR bits during a host read of the MSR as follows:

Signal	MCR Bit	MSR Bit
DTR	MCR0	MSR5
RTS	MCR1	MSR4
Out1	MCR2	MSR6
Out2	MCR3	MSR7

0 = Disables loop.

Bit 5-7: Not Used. (Always 0).

3.13.10. Line Status Register (LSR)

The host can read and write all bits in this register, however it is recommended that the host not write to the LSR register. The CPU can read all bits but can only write to LSR1 (Overrun Error).

The typical use of these bits in a 16450/16550A interface application is:

Bit 0: **Receiver Data Ready (DR).** This bit is cleared by MCU reset. DR is set by hardware whenever the RX UART simulator pointer, RUPTR, goes from empty to not empty. Cleared when the RUPTR pointer indicates empty. When Freeze (FSR5) is set, LSR0 is no longer controlled by the state of the RUPTR. When FSR5 = 1, the DR bit can be cleared by reading the RX FIFO Buffer.

Bit 1: **Overrun Error (OE).** This bit is cleared by RX FIFO reset (FCR1 = 1) or MCU reset. OE is set or reset by CPU write. This bit is reset when the host reads LSR. This bit is cleared by MCU reset.

Bit 2-4: **Receive Line Status**. These bits are cleared by RX FIFO reset (FCR1 = 1) or MCU reset. They are updated automatically from the error field of RX FIFO following the first CPU write or each host read to the RX FIFO. LSR bits 2, 3, 4 and 7 and the RX FIFO output data latch are all updated at this time. LSR error data must be read prior to reading the RX FIFO output or the error data is lost. These bits are cleared to zero when the host reads LSR. These bits are cleared by MCU reset.

Bit 2: **Parity Error (PE).** See bits 2-4.

Bit 3: **Framing Error (FE).** See bits 2-4.

Bit 4: **Break Interrupt (BI)**. See bits 2-4.

Bit 5: **TX FIFO Buffer Empty (THRE)**. This bit is set by MCU reset. THRE is set by hardware each time the transmitter UART pointer, TUPTR, transitions from not empty to empty. It is reset by a host write to the TX FIFO. If a HINT interrupt occurs and IIR identifies THRE as the source, the interrupt is cleared while THRE remains set. Whenever the host writes to the Interrupt Enable Register enabling a THRE interrupt and THRE is set, a THRE interrupt is regenerated.

Bit 6: **Transmitter Underrun (TEMT)**. Set by MCU reset or by hardware when the TX FIFO has been empty for at least one TUCLK time. Reset when the host writes to the TX FIFO.

Bit 7: **RX FIFO Error**. Set by MCU hardware if at least one error bit is set in the unread error field of the RX FIFO. Reads zero in the 16450 mode.

3.13.11. Modem Status Register (MSR)

All Modem Status Register (MSR) bits can be read or written to by either the host or the CPU. The CPU can write all bits, however the host can only write to bits 0-3. Bits 0-3 are set to a 1 by the CPU writing a 0 to the particular bit. Bits 0-3 are unaffected when the MCU writes a 1 to a bit position. Bits 0-3 are cleared when the host reads the MSR. It is recommended that the host not write to the MSR. A host MSR read is modified when Loop is selected (MCR4 = 1). Bits 0-3 are cleared by MCU reset.

When Loop is not selected (MCR4 = 0), a host MSR read generates the following data:

(MSR7/MSR6/MSR5/MSR4/MSR3/MSR2/MSR1/MSR0).

When Loop is selected (MCR4 = 1), a host MSR read generates the following data:

(MCR3/MCR2/MCR0/MCR1/MSR3/MSR2/MSR1/MSR0).

The typical use of these bits in a 16450/16550A interface application is:

The Modem Status Register (MSR) reports current state and change information of the modem. Bits 4-7 supply current state and bits 0-3 supply change information. The change bits are set to a logic 1 whenever a control input from the modem changes state from the last MSR read by the host. Bits 0-3 are reset to logic 0 when the host reads the MSR or upon reset. Whenever bits 0, 1, 2, or 3 are set to a logic 1, a Modem Status Interrupt is generated.

Bit 0: **Delta Clear to Send (DCTS).** This bit is set to a logic 1 when the CTS bit has changed since the MSR was last read by the host.

Bit 1: **Delta Data Set Ready (DDSR).** This bit is set to a logic 1 when the DSR bit has changed since the MSR was last read by the host.

Bit 2: **Trailing Edge of Ring Indicator (TERI).** This bit is set to a logic 1 when the RI bit changes from a 1 to a 0 state since the MSR was last read by the host.

Bit 3: **Delta Data Carrier Detect (DDCD).** This bit is set to a logic 1 when the DCD bit changes state since the MSR was last read by the host.

Bit 4: **Clear to Send (CTS).** This bit indicates the logic state of the CTS output. If Loopback is selected (MCR4 = 1), this bit reflects the state of RTS in the MCR (MCR1).

Bit 5: **Data Set Ready (DSR).** This bit indicates the logic state of the DSR output. If Loopback is selected (MCR4 = 1), this bit reflects the state of DTR in the MCR (MCR0).

Bit 6. **Ring Indicator (RI).** This bit indicates the logic state of the RI output. If Loopback is selected (MCR4 = 1), this bit reflects the state of OUT1 in the MCR (MCR2).

Bit 7. **Data Carrier Detect (DCD).** This bit indicates the logic state of the DCD output. If Loopback is selected (MCR4 = 1), this bit reflects the state of OUT2 in the MCR (MCR3).

3.13.12. Divisor Latch Registers

The Divisor Latch LSB (least significant byte) and Divisor Latch MSB (most significant byte) are two read-write registers. HCR bit 4 is set automatically whenever the host writes to either divisor latch byte, thus providing a flag to the CPU that the Divisor Latch has been updated. User-provided software must read and interpret the Divisor Latch contents.

All bits in both registers can be read or written to by either the host or the CPU. This register is not initialized by MCU reset.

3.13.13. Scratch Register

All Scratch Register bits can be read or written to by either the host or the CPU. There are no flag bits associated with this register. This register is not initialized by MCU reset.

3.13.14. FIFO Control Register (FCR)

The host can write all bits but cannot read this register. The CPU can read all bits but cannot write to this register. Bits 1, 2, 4 and 5 read high to the CPU.

The typical use of these bits in a 16450/16550A interface application is:

Bit 0: **FIFO Enable.** Control bit.

1 = FIFO mode (16550A mode) is selected and both FIFOs are enabled.

0 = 16450 mode is selected and all bits are cleared in both FIFOs.

Bit 1: **RX FIFO Reset.** Control bit.

1 = All bytes in the RX FIFO are cleared. This bit is cleared automatically by the CPU.

Bit 2: **TX FIFO Reset.** Control bit.

1 = All bytes in the TX FIFO are cleared. This bit is cleared automatically by the CPU.

Bit 3: **DMA Mode Select.** Control bit. Selects or deselects the DMA mode when FIFO mode is selected (FCR0 = 1).

1 = Selects the DMA operation when FIFO mode is selected (FCR0 = 1).

0 = Selects non-DMA operation when FIFO mode is selected (FCR0 = 1). Non-DMA operation is also selected when 16450 mode is selected (FCR0 = 0).

DMA operation in FIFO mode. RXRDY is asserted when the number of characters in the RX FIFO exceeds the value in the RX FIFO Trigger Level (FCR6-7) or the RXTO time-out has occurred. RXRDY will go inactive when there are no more characters in the RX FIFO.

TXRDY is asserted when there one or more unfilled locations in the TX FIFO. TXRDY goes inactive when the TX FIFO is completely full.

Non-DMA operation in FIFO mode. RXRDY is asserted when there are one or more characters in the RX FIFO. RXRDY goes inactive when there are no more characters in the RX FIFO.

TXRDY is asserted when there are no characters in the TX FIFO. TXRDY goes inactive when the first character is loaded into the TX FIFO Buffer.

Bits 4-5: **Not used.**

Bits 6-7: **Receiver FIFO Trigger Level.** FCR7 and FCR6 set the trigger level for the RX FIFO interrupt.

FCR7	FCR6	RX FIFO Trigger Level (Bytes)
0	0	01
0	1	04
1	0	08
1	1	14

3.13.15. FIFO Interrupt Enable Register (FIER)

The CPU can read and write all bits. The host cannot access this register. The FIER is cleared by MCU reset.

Bit 0: **RCHE Interrupt Enable.** This bit is set and cleared by a CPU write to the FIER.

Bit 1: **RCEMT Interrupt Enable.** This bit is set and cleared by a CPU write to the FIER.

Bit 2: **RX FIFO Reset.** This bit is set and cleared by a CPU write to the FIER. When set, FIER2 clears the RXPTR pointer, the RUPTR pointer and the RX time-out counter. FIER2 automatically resets itself to zero.

Bit 3: **TX FIFO Reset.** This bit is set and cleared by a CPU write to the FIER. When set, FIER3 clears the TXPTR pointer, the TUPTR pointer and the TX time-out counter. FIER3 automatically resets itself to zero.

Bit 4: **RUCLK Off.** When FIER4 is set, IWRX rather than RUCLK is used to increment RUPTR (see Figure 3-21). As a result, when set, the RXPTR and RUPTR become locked together. In the GP Mode only, setting FIER4 causes both the TXPTR and TUPTR to be decremented by a CPU FIFO read. As a result, the TXPTR and TUPTR are also locked together. This bit is set and cleared by a CPU write to the FIER.

Bit 5: **UART Timing Select.** When FIER5 is set, RUCLK and TUCLK timing is derived from PTGB. When XFS5 is reset RUCLK and TUCLK timing is derived from TIMA (see Figure 3-20) This bit is set and cleared by a CPU write to the FIER.

Bit 6: **TCDA Interrupt Enable.** This bit is set and cleared by a CPU write to the FIER. When this bit is a 1 and TCDA (FSR6) is also a 1, IRQ3 is asserted.

Bit 7: **TCHF Interrupt Enable.** This bit is set and cleared by a CPU write to the FIER. When this bit is a 1 and TCHF (FSR7) is a 1, IRQ3 is asserted.

3.13.16. FIFO Status Register (FSR)

The CPU can read all bits but can only write to bits 2 through 5. The host cannot access this register. Except for bit 0 which initializes high, all bits are initialized to zero by MCU reset. Bits 2 - 4 are inputs to the error field of the RX FIFO.

Bit 0: **RX FIFO Half Empty Flag (RCHE).** This bit is set and cleared by hardware. It provides RX FIFO status to the CPU. RCHE is set whenever the RX FIFO contains between 0 and 7 unread bytes of data. If FIER0 and FSR0 are both set, IRQ3 is asserted.

Bit 1: **RX FIFO Empty Flag (RCEMT).** This bit is set and cleared by hardware. It provides RX FIFO status to the MCU CPU. RCEMT is set whenever the RX FIFO is empty. If FIER1 and FSR1 are both set, IRQ3 is asserted.

Bit 2: **Receiver Parity Error (PE).** This bit is set and cleared by a CPU write to the FSR.

Bit 3: **Receiver Framing Error (FE).** This bit is set and cleared by a CPU write to the FSR.

Bit 4: **Receiver Break Interrupt (BI).** This bit is set and cleared by a CPU write to the FSR.

Bit 5: **DR Freeze (DRF).** Freeze halts the assertion of DR by RUPTR logic. Freeze is only used in the 16450 mode. If DR is asserted while Freeze is true, DR is cleared by a host read of the RX FIFO Buffer. This bit is set and cleared by a CPU write to the FSR.

Bit 6: **TX FIFO Data Available Flag (TCDA).** This bit is set and cleared by hardware. It provides TX FIFO status to the CPU. It is true whenever the TX FIFO contains between 1 and 16 unread bytes of data. If FIER6 and FSR6 are both set, IRQ3 is asserted.

Bit 7: **TX FIFO Half Full Flag (TCHF).** This bit is set and cleared by hardware. It provides TX FIFO status to the CPU. It is true whenever the TX FIFO contains between 8 and 16 unread bytes of data. If FIER7 and FSR7 are both set, IRQ3 is asserted.

3.13.17. GP FIFO Status Register (GPFS)

All bits can be read by either the host or the CPU (see Table 3-22). The host cannot access this register in either the 16450 or 16550A mode. For the C40 and L39, in the 16550 mode only, the MCU can write to bit 4. The host can write to bits 2, 3, 4 and 7 only. MCU reset initializes bits 0-4 and bit 7 to 0 and initializes bits 5 and 6 to 1.

Bit 0: **RX FIFO Data Available.** When operating in the GP FIFO mode, GPFS0 = 1 indicates the not empty status of RUPTR. When operating in the 16450 or 16550 mode, GPFS0 = 1 indicates that the RX FIFO is full.

Bit 1: **RX FIFO Trigger Level Flag.** GPFS1 = 1 indicates that the RUPTR has reached or exceeds the trigger level specified by RTL0 and RTL1.

Bit 2: **RX FIFO Trigger Level Interrupt Enable.** GPFS2 = 1 enables an HINT interrupt to be automatically generated whenever GPFS1 is set.

Bit 3 - 4: **RUPTR Trigger Level (RTL0 and RTL1).** GPFS3 and GPFS4 set the trigger level for the RX FIFO interrupt.

GPFS4	GPFS3	RUPTR Trigger Level (Bytes)
0	0	01
0	1	04
1	0	08
1	1	14

Bit 3: **16550 Mode TUCLK OFF.** When bit 4 is set in the 16550 mode, the TUPTR is decremented by IRTX rather then TUCLK. As a result the TUPTR becomes locked to the TXPTR value.

Bit 5: **Tx FIFO Half Empty.** GPFS5 is a 1 when there are 0 to 7 data bytes in the TX FIFO.

Bit 6: **Tx FIFO Empty Flag.** GPFS6 is a 1 when there are no data bytes in the TX FIFO.

Bit 7: **Tx FIFO Empty Interrupt Enable.** GPFS7 = 1 enables an HINT interrupt to be automatically generated whenever GPRS6 is a 1.

3.13.18. Host Handshake Register (HHR)

The Host Handshake Register (HHR) supports a defined handshake protocol between the host and the CPU (Table 3-22). All bits can be read by either the host or the CPU. The host cannot access this register in 16450 or 16550A mode. Bits 0-3 are interrupt enable bits and bits 4-7 are interrupt flag bits. All bits are cleared by MCU reset.

Bit 0: **Host Attention Flag 2 Interrupt Enable (HE2).** This bit, when set to a 1, enables HINT output to be asserted when the Host Attention Flag 2 (HHR4) is set. When this bit is reset to a 0, HINT will not be asserted based on HHR4. This bit can be written only by the host.

Bit 1: **Host Attention Flag 1 Interrupt Enable (HE1).** This bit, when set to a 1, enables HINT output to be asserted when the Host Attention Flag 1 (HHR5) is set. When this bit is reset to a 0, HINT will not be asserted based on HHR5. This bit can be written only by the host.

Bit 2: **Controller Attention Flag 2 Interrupt Enable (CE2).** This bit, when set to a 1, enables IRQ3 to be asserted when the CPU Attention Flag 2 (HHR6) is set. When this bit is reset to a 0, IRQ3 will not be asserted based on HHR6. This bit can be written only by the CPU.

Bit 3: **Controller Attention Flag 1 Interrupt Enable (CE1).** This bit, when set to a 1, enables IRQ3 to be asserted when the CPU Attention Flag 1 (HHR7) is set. When this bit is reset to a 0, IRQ3 will not be asserted based on HHR7. This bit can be written only by the CPU.

Bit 4: **Host Attention Flag 2 (HF2).** This is an attention bit from the CPU to the host. This bit can be set only by the CPU writing a 0 and can be cleared only by the host writing a 0. HF2 is not affected by the CPU writing a 1 or the host writing a 1.

Bit 5: **Host Attention Flag 1. (HF1)** This is an attention bit from the CPU to the host. This bit can be set only by the CPU writing a 0 and can be cleared only by the host writing a 0. HF1 is not affected by the CPU writing a 1 or the host writing a 1.

Bit 6: **Controller Attention Flag 2 (CF2).** This is an attention bit from the host to the CPU. This bit can be set only by the host writing a 1 and can be cleared only by the CPU writing a 0. CF2 is not affected by the host writing a 0 or the CPU writing a 1.

Bit 7: **Controller Attention Flag 1 (CF1).** This is an attention bit from the host to the CPU. This bit can be set only by the host writing a 1 and can be cleared only by the CPU writing a 0. CF1 is not affected by the host writing a 0 or the CPU writing a 1.

3.13.19. 16550 DMA Operation

Transferring Received Data from the RX FIFO to the PC Bus Memory

The PC bus (host bus) operation to transfer a received data byte from the MCU I/O (RX FIFO) to PC bus memory is as follows:

1. Select 16550 host interface mode and ensure that the USART serial mode is not selected (SM7 = 0). Note that the RXRDY is available as an output signal only in the 16550 host interface mode when USART serial mode is not selected.

2. The MCU sets RXRDY high when the RX FIFO conditions meet the FIFO Control Register specified limit. This requests a DMA transfer on the channel to which the RXRDY line is connected.

3. The PC bus responds by setting DACK (MCU HRACKP signal) low to indicate that the DMA controller now is controlling the PC data bus.

4. The DMA controller places the memory address location on the address bus to which the data will be transferred and sets the I/O READ and memory WRITE signals low. The MCU places the receive FIFO data byte to be transferred to memory on the data bus. In this mode the MCU ignores the host address bus (assumes it is $00) and host CHIP ENABLE (assumes it is enabled).

5. The DMA controller sets memory WRITE and I/O READ lines high transferring the byte from the MCU I/O (RX FIFO) to PC bus memory in one cycle.

6. The DMA controller sets DACK high and relinquishes the PC bus to the host controller. Dependent upon the FIFO conditions and the FIFO Control Register specified limit, the MCU either initiates another byte transfer immediately by continuing to hold RXRDY high, or sets RXRDY low and waits until the RX FIFO meets the limit to initiate a transfer.

Transferring Transmit Data from PC Bus Memory to the TX FIFO

The PC bus (host bus) operation to transfer a transmit data byte from PC bus memory to the MCU I/O (TX FIFO) is as follows:

1. Select 16550 host interface mode and ensure that the USART serial mode is not selected (SM7 = 0). Note that the TXRDY is available as an output signal only in the 16550 host interface mode when USART serial mode is not selected.

2. When the TX FIFO is empty or has one empty location, the MCU sets TXRDY high. This requests a DMA transfer on the channel to which the TXRDY line is connected.

3. The PC bus responds by setting DACK (MCU HTACKP signal) low to indicate that the DMA controller is now controlling the PC bus.

4. The memory address location of the data byte to be transmitted is placed on the address bus by the DMA controller. The DMA controller sets the I/0 WRITE and the memory READ lines low. In this mode, the MCU ignores the host address bus (assumes it is $00) and host CHIP ENABLE (assumes it is enabled). The memory device places the data byte to be transmitted on the PC data bus.

5. The DMA controller sets memory READ and the I/O WRITE lines high. The MCU inputs the data from the data bus so the byte is transferred from the PC bus memory to the MCU I/O (TX FIFO) in one cycle.

6. The DMA controller sets DACK high and relinquishes the PC bus to the host controller. If the TX FIFO is has at least one empty location, dependent on the FIFO Control Register setup, the MCU continues to hold the TXRDY line high in order to initiate another DMA transfer, or if not sets the TXRDY line low.

3.14. EXPANSION BUS

The expansion bus extends internal address, data and control bus lines outside the MCU. This allows the MCU to operate as a microprocessor by interfacing with external memory and/or other peripheral devices. The expansion bus waveforms and timing are described in Section 5.

3.15. TEST MODE

The Test mode is selected by applying a low voltage to the TSTP pin.

in the Test mode, the internal ROM is deactivated and the expansion bus activated when ROM addresses are selected. Reads from addresses $000C-$000F are mapped to the expansion port.

Internal reads from Page 0, 1 and 2 are mapped externally on the expansion bus whenever the TSTP pin is active. This provides for monitoring of internal read operations.

3.16. MASK OPTIONS

3.16.1. Selectable Options

The mask options are selectable upon production part order.

3.16.2. Mask Option Register (MOR)

The Mask Option Register (MOR) at location $0008 reports the selected mask options (Table 3-25).

Table 3-25. Register Bit Assignments: Mask Option Register - 0008h

Addr.	Function	Bit							
		7	6	5	4	3	2	1	0
0008	Mask Option Register (MOR)	Mask Option Bit 7 (MOR7)	Mask Option Bit 6 (MOR6)	Mask Option Bit 5 (MOR5)	Mask Option Bit 4 (MOR4)	Mask Option Bit 3 (MOR3)	Mask Option Bit 2 (MOR2)	Mask Option Bit 1 (MOR1)	Mask Option Bit 0 (MOR 0)

Bit 0: **Mask Option Bit 0 (MOR0).** Product specific.

Bit 1: **Mask Option Bit 1 (MOR1).** Product specific.

Bit 2: **Mask Option Bit 2 (MOR2).** Product specific.

Bit 3: **Mask Option Bit 3 (MOR3).** Product specific.

Bit 4: **Mask Option Bit 4 (MOR4).** Product specific.

Bit 5: **Mask Option Bit 5 (MOR5).** Product specific.

Bit 6: **Mask Option Bit 6 (MOR6).** Product specific.

Bit 7: **Mask Option Bit 7 (MOR7).** Product specific.

3.17 CYCLIC REDUNDANCY CHECK (CRC) HARDWARE

The MCU supports the standard 16-bit CRC polynomial algorithm:

$$CRC16 = (x^{16} + x^{12} + x^5 + 1)$$

CRC16 polynomial check = $F0B8.

The operation of the CRC is illustrated in Figure 3-22.

The CRC polynomial is initialized to $FFFF by writing to address $05FF. The parallel in serial out shift register is loaded with data by a CPU write to address $05FE. A minimum of eight clock cycles must be allowed between CPU writes. The resulting polynomial can be read at any time after allowing for the shifting operation to complete. The LSB part of the polynomial is read at address $05FE and the MSB part at address $05FF.

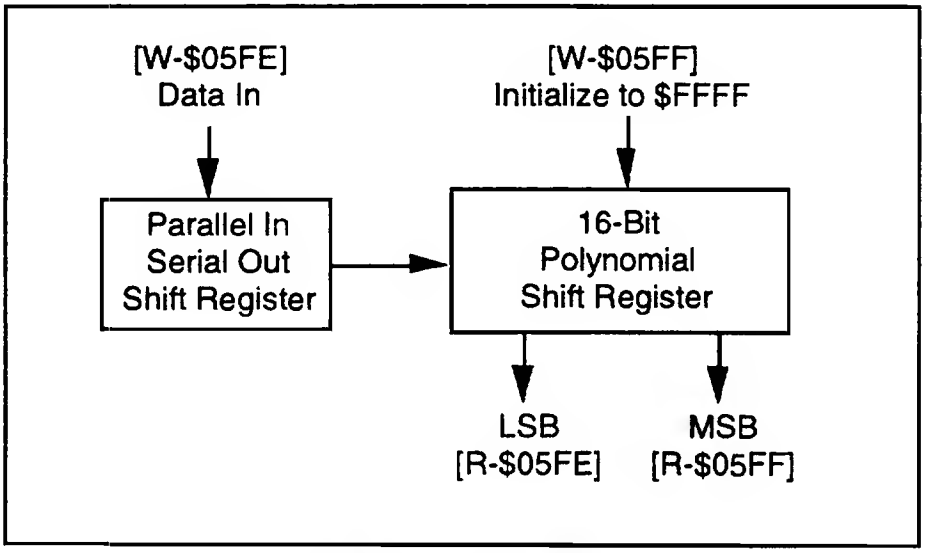


Figure 3-22. CRC Operation